

明 細 書

マルチプロセッサシステムおよびメモリアクセス処理方法

技術分野

本発明は、メモリを共有する密結合型のマルチプロセッサシステムに関し、特に入出力デバイスからのメモリアクセスを処理するマルチプロセッサシステムおよびメモリアクセス処理方法に関する。

背景技術

PCIバス仕様リビジョン2.1等の規定では、PCIバス上の入出力デバイスを要求元にしたライトメッセージは順序を保障しなければならないという制約がある。即ち、先行のライトメッセージが完了してから後続のライトメッセージが完了することを保障しなければならない。

図1はディレクトリ方式によってデータの一貫性を維持するマルチプロセッサシステムの構成を示す図である。マルチプロセッサシステムは、複数のプロセッサノード101-1、…、101-i、…、101-m (i、mは2以上の整数)と、複数の入出力ノード103-1、…、103-j、…、103-n (j、nは2以上の整数)とを具備している。複数のプロセッサノード101-1～101-mと複数の入出力ノード103-1～103-nとは、ネットワーク102に接続され、外部からのクロックに応じて動作する。プロセッサノード101-1は、プロセッサ110-1-1、110-1-2と、ディレクトリ120-1と、主記憶部(メモリ)130-1と、メモリコントローラ140-1とを含む。これは、他のプロセッサノードも同じである。各主記憶部には複数のデータが格納されている。複数のデータの各々は、その内容を表す値を含んでいる。一方、入出力ノード103-1は、入出力コントローラ150-1と、外部からの命令によりメッセージを発行する複数の入出力デバイス160-1-1、160-1-2とを含む。これは、他の入出力ノードも同じである。なお、以下では、便宜上、図1に示されていない参照番号を使用して説明を行うことがある。例えば、

プロセッサノード101-2は図示されていないが、主記憶部130-2、メモリコントローラ140-2等を含む。

図1に示すようなマルチプロセッサシステムにおいて前述のライトメッセージの順序制約を満たす従来技術を紹介する。特開2001-216259号公報(以下、特許文献1と呼ぶ)の「発明が解決しようとする課題」には、先行するライトメッセージの完了が保障されるまで後続するライトメッセージの発行を留めることで順序制約を満たす技術(以降ではこれを従来技術1と呼ぶ)が開示されている。

図1、図2を参照しながら、順序制約のあるライトメッセージが連続して発行された場合について従来技術1の動作を説明する。図2は、入出力デバイス160-1-1が、データA、B、Cに対するライトメッセージをそれぞれステップ1、2、3で発行した場合の動作を示している。ここで、データAおよびデータBはプロセッサノード101-1をホームとし、データCはプロセッサノード101-2(図示せず)をホームとするデータであるとする。また、1ステップは1クロックに対応する。

入出力コントローラ150-1は、ステップ2にて、入出力デバイス160-1-1からのライトメッセージとしてライトAメッセージを受け取る。入出力コントローラ150-1は、ステップ3にて、ライトAメッセージで指定される値を含む更新Aメッセージをホームのメモリコントローラ140-1に宛ててネットワーク102に出力する。

メモリコントローラ140-1は、ステップ4にて、入出力コントローラ150-1からの更新Aメッセージを受けて、主記憶部130-1に格納されたデータの値を更新Aメッセージで指定される値に更新する。メモリコントローラ140-1は、ステップ5にて、入出力コントローラ150-1に宛ててネットワーク102に完了Aメッセージを出力する。入出力コントローラ150-1は、ステップ6にて、メモリコントローラ140-1からの完了Aメッセージを受け取り、先行するライトAが完了したことを認識する。

入出力コントローラ150-1はまた、ステップ3にて、第2ライトメッセージとしてライトBメッセージを受け取る。この場合、入出力コントローラ150

ー1は、先行するライトAの完了するステップ6までライトBメッセージを留め置く。入出力コントローラ150-1は、ステップ7にて、ライトBメッセージで指定される値を含む更新Bメッセージをメモリコントローラ140-1に宛ててネットワーク102に出力する。

メモリコントローラ140-1は、ステップ8にて、入出力コントローラ150-1からの更新Bメッセージを受けて、主記憶部130-1に格納されたデータの値を更新Bメッセージで指定される値に更新する。メモリコントローラ140-1は、ステップ9にて、入出力コントローラ150-1に宛ててネットワーク102に完了Bメッセージを出力する。入出力コントローラ150-1は、ステップ10にて、メモリコントローラ140-1からの完了Bメッセージを受け取り、先行するライトBが完了したことを認識する。

入出力コントローラ150-1は更に、ステップ4にて、第3ライトメッセージとしてライトCメッセージを受け取る。この場合、入出力コントローラ150-1は、先行するライトAおよびライトBが両方とも完了するステップ10までライトCメッセージを留め置く。入出力コントローラ150-1は、ステップ11にて、ライトCメッセージで指定される値を含む更新Cメッセージをメモリコントローラ140-2（図示せず）に宛ててネットワーク102に出力する。

メモリコントローラ140-2は、ステップ12にて、入出力コントローラ150-1からの更新Cメッセージを受けて、主記憶部130-2（図示せず）に格納されたデータの値を更新Cメッセージで指定される値に更新する。メモリコントローラ140-2は、ステップ13にて、入出力コントローラ150-1に宛ててネットワーク102に完了Cメッセージを出力する。入出力コントローラ150-1は、ステップ14にて、メモリコントローラ140-2からの完了Cメッセージを受け取り、ライトCが完了したことを認識する。

このように、入出力コントローラ150-1は、先行するライトメッセージが完了してから次のライトメッセージを発行することで、入出力デバイス160-1-1、160-1-2が発行した複数のライトメッセージの順序を保障することができる。しかし、3つのライトメッセージを処理するのに14ステップを要する。これは、他の入出力コントローラについても同様である。

特許文献1にはまた、ライトメッセージの処理時間が長く、性能が劣化する問題を解決する技術（以降ではこれを従来技術2と呼ぶ）が開示されている。従来技術2は、同一プロセッサノードを宛先とするライトメッセージの連続発行、即ち先行するメッセージの完了が保障される前に後続するメッセージを発行することを可能とする。

図1、図3を参照しながら、従来技術2の動作を説明する。

入出力コントローラ150-1は、ステップ2にて、第1ライトメッセージとしてライトAメッセージを受け取り、ライトAメッセージがプロセッサノード101-1（メモリコントローラ140-1）をホームとするライトであることを認識する。入出力コントローラ150-1は、ステップ3にて、ライトAメッセージで指定される値を含む更新Aメッセージをホームのメモリコントローラ140-1に宛ててネットワーク102に出力する。

メモリコントローラ140-1は、ステップ4にて、入出力コントローラ150-1からの更新Aメッセージを受けて主記憶部130-1のデータを更新する。メモリコントローラ140-1は、ステップ5にて、入出力コントローラ150-1に宛ててネットワーク102に完了Aメッセージを出力する。入出力コントローラ150-1は、ステップ6にて、メモリコントローラ140-1からの完了Aメッセージを受け取り、先行するライトAが完了したことを認識する。

入出力コントローラ150-1はまた、ステップ3にて、第2ライトメッセージとしてライトBメッセージを受け取る。入出力コントローラ150-1は、ライトBメッセージがプロセッサノード101-1（メモリコントローラ140-1）をホームとするライトであることを認識し、先行するライトAメッセージと同じホームであることを認識する。この場合、入出力コントローラ150-1は、ライトAの完了を待つことなく、ステップ4にて、ライトBメッセージで指定される値を含む更新Bメッセージをホームのメモリコントローラ140-1に宛ててネットワーク102に出力する。

メモリコントローラ140-1は、ステップ5にて、入出力コントローラ150-1からの更新Bメッセージを受けて主記憶部130-1のデータを更新する。メモリコントローラ140-1は、ステップ6にて、入出力コントローラ150-

ー1に宛ててネットワーク102に完了Bメッセージを出力する。入出力コントローラ150-1は、ステップ7にて、メモリコントローラ140-1からの完了Bメッセージを受け取る。

入出力コントローラ150-1は更に、ステップ4にて、第3ライトメッセージとしてライトCメッセージを受け取る。入出力コントローラ150-1は、ライトCメッセージがプロセッサノード101-2（メモリコントローラ140-2）（いずれも図示せず）をホームとするライトであることを認識し、先行するライトAメッセージ、ライトBメッセージとはホームが異なることを認識する。この場合、入出力コントローラ150-1は、両ライトメッセージ（ライトAメッセージ、ライトBメッセージ）とも完了するステップ7までライトCメッセージを留め置く。入出力コントローラ150-1は、ステップ8にて、ライトCメッセージで指定される値を含む更新Cメッセージをメモリコントローラ140-2に宛ててネットワーク102に出力する。

メモリコントローラ140-2は、ステップ9にて、入出力コントローラ150-1からの更新Cメッセージを受けて主記憶部130-2のデータを更新する。メモリコントローラ140-2は、ステップ10にて、入出力コントローラ150-1に宛ててネットワーク102に完了Cメッセージを出力する。入出力コントローラ150-1は、ステップ11にて、メモリコントローラ140-1からの完了Cメッセージを受け取り、ライトCが完了したことを認識する。

このように、入出力コントローラ150-1は、ホームを同じにするライトメッセージは連続して発行し、ホームが異なるライトメッセージは先行するライトメッセージが完了してから発行する。ネットワーク102は2点間のメッセージの順序を保障するので、上記例ではライトAメッセージとライトBメッセージはその順でメモリコントローラ140-1に到着することが保障される。そのため、ライトAメッセージを追い越してライトBメッセージが先にメモリコントローラ140-1で処理されることはなく順序を保障することができる。

しかし、この従来技術2でも3つのライトメッセージを処理するのに11ステップを要する。

従来技術1、従来技術2では、入出力コントローラが、異なるプロセッサノード

ドを宛先とする入出力デバイスからの複数のライトメッセージを連続して処理することができない。このため、入出力コントローラがライトメッセージの処理を行う場合に要する時間は、長くなってしまう。

本発明の課題は、入出力コントローラが、異なるプロセッサノードを宛先とする入出力デバイスからの複数のライトメッセージを連続して処理することができるマルチプロセッサシステムを提供することにある。

本発明の他の課題は、入出力コントローラがライトメッセージの処理を行う場合に要する時間を短くすることができるマルチプロセッサシステムを提供することにある。

発明の開示

本発明によるプロセッサシステムは、ネットワークに接続された複数の資源ノードとネットワークに接続された複数の制御ノードと複数の制御コントローラとを具備する。

複数の資源ノードはそれぞれ、複数の資源を保持する資源保持部と、複数の資源がフリー状態にあるかロック状態にあるかを管理する資源情報保持部とを含む。

複数の制御ノードはそれぞれ、資源に対する要求を複数発行する複数の制御デバイスを含む。

複数の要求を発行した制御デバイスを含む制御ノードは複数の要求を対応する制御コントローラに転送する。

要求を受けた少なくとも1つの制御コントローラは、該要求が対象とする資源を保持する資源ノードに要求メッセージを発行する。

要求メッセージを受けた資源ノードは、対象とする資源の状態がフリー状態にあることを確認して資源の状態をロック状態に変更し、続いて許可メッセージを少なくとも1つの制御コントローラに発行する。

許可メッセージを受けた少なくとも1つの制御コントローラは、当該少なくとも1つの制御コントローラが受けた要求に先行する他の複数の要求について全て許可メッセージを受け取り済みであることを確認して、要求が対象としている資源を保持する資源ノードに更新メッセージを発行する。

更新メッセージを受けた資源ノードは、対象とする資源の状態をフリー状態に変更する。

本発明によるプロセッサシステムにおいては、複数の制御コントローラがそれぞれ、複数の制御ノード内に備えられても良い。この場合、各制御ノードの制御コントローラは、各制御ノードの複数の制御デバイスによって発行される複数の要求を調停するセレクトを含むことが望ましい。

本発明によるプロセッサシステムにおいてはまた、複数の制御コントローラがネットワークに接続されても良い。この場合、各制御ノードは、複数の制御ノードの各々の複数の制御デバイスによって発行される複数の要求を調停するセレクトを含むことが望ましい。

本発明によるプロセッサシステムにおいては更に、複数の制御コントローラがそれぞれ、複数の資源ノード内に備えられても良い。この場合、各制御ノードは、複数の制御ノードの各々の複数の制御デバイスによって発行される複数の要求を調停するセレクトを含むことが望ましい。

本発明によるプロセッサシステムにおいては更に、複数の資源ノードと複数の制御ノードとが、それぞれ複数のノードを構成するようにされても良い。この場合、各ノードは、複数のノードの各々の複数の制御デバイスによって発行される複数の要求を調停するセレクトを含むことが望ましい。

本発明によるプロセッサシステムにおいては、各制御コントローラが開放処理部を具備しても良い。この場合、要求メッセージを受けた資源ノードは、対象とする資源の状態がロック状態にあった場合、開放要求メッセージを少なくとも1つの制御コントローラに発行する。

開放要求メッセージを受けた少なくとも1つの制御コントローラの開放処理部は開放処理を開始する。

開放処理は、該要求が対象としている資源を保持する資源ノードに要求メッセージを発行する処理と、新規要求の受付を停止する処理と、該少なくとも1つの制御コントローラが受けた要求に後続する他の複数の後続要求の各々について、既に許可メッセージを受け取っていれば、あるいは開放処理中に許可メッセージを受け取れば、該後続要求が対象としている資源を保持する資源ノードに開放メ

ッセージを発行する処理とを含む。

開放要求メッセージの要求に対する許可メッセージを受け取った少なくとも1つの制御コントローラは開放処理を終了する。

本発明によるプロセッサシステムにおいてはまた、資源情報保持部が管理する状態は、割り込み可ロック状態とリクエストロック状態とを含んでも良い。この場合、要求を受けた少なくとも1つの制御コントローラは、該少なくとも1つの制御コントローラが受けた要求に先行する他の複数の要求の全てについて許可メッセージを受け取り済みでなければ弱要求メッセージを発行する。

弱要求メッセージを受けた資源ノードは、対象とする資源の状態がフリー状態にあることを確認して、資源の状態を割り込み可ロック状態に変更し、続いて許可メッセージを少なくとも1つの制御コントローラに発行する。

割り込み可ロック状態は、少なくとも1つの制御コントローラを特定する情報を含んでも良い。

要求メッセージを受けた資源ノードは、対象とする資源の状態がロック状態にあった場合、不許可メッセージを少なくとも1つの制御コントローラに発行する。

弱要求メッセージを受けた資源ノードは、対象とする資源の状態がロック状態あるいは割り込み可ロック状態にあった場合、不許可メッセージを少なくとも1つの制御コントローラに発行する。

不許可メッセージを受けた少なくとも1つの制御コントローラは、該少なくとも1つの制御コントローラが受けた要求に先行する他の複数の要求の全てについて許可メッセージを受け取り済みであれば要求メッセージを、受け取り済みでなければ弱要求メッセージを発行する。

要求メッセージを受けた資源ノードは、対象とする資源の状態が割り込み可ロック状態にあった場合、資源の状態をリクエストロック状態に変更し、続いて割り込み可ロック状態に含まれる情報が示す制御コントローラに宛てて再試行要求メッセージを出力する。

再試行要求メッセージを受け取った制御コントローラは、再試行処理を行う。

再試行処理は、該資源に対する要求を特定する処理と、該要求が対象としている資源を保持する資源ノードに開放メッセージを発行する処理と、特定した要求

がまだ更新メッセージを発行していなければ、該要求について許可メッセージをまだ受け取っていない状態に変更し、該制御コントローラが受けた要求に先行する他の複数の要求の全てについて許可メッセージを受け取り済みであれば要求メッセージを、受け取り済みでなければ弱要求メッセージを発行する処理とを含む。

開放メッセージを受けた資源ノードは、資源の状態をロック状態に変更し、続いて許可メッセージを少なくとも1つの制御コントローラに発行する。

本発明によるアクセス処理方法は、ネットワークに接続された複数の資源ノードとネットワークに接続された複数の制御ノードと複数の制御コントローラとを具備するプロセッサシステムであって、複数の資源ノードがそれぞれ、複数の資源を保持する資源保持部と、複数の資源がフリー状態にあるかロック状態にあるかを管理する資源情報保持部とを含み、複数の制御ノードがそれぞれ、資源に対する要求を複数発行する複数の制御デバイスを含む、プロセッサシステムに適用される。

本発明によるアクセス処理方法は、複数の要求を対応する少なくとも1つの制御コントローラに転送するステップと、該要求が対象とする資源を保持する資源ノードに要求メッセージを発行するステップと、対象とする資源の状態がフリー状態にあることを確認して資源の状態をロック状態に変更し、続いて許可メッセージを少なくとも1つの制御コントローラに発行するステップと、当該少なくとも1つの制御コントローラが受けた要求に先行する他の複数の要求について全て許可メッセージを受け取り済みであることを確認して、要求が対象としている資源を保持する資源ノードに更新メッセージを発行するステップと、対象とする資源の状態をフリー状態に変更するステップとを含む。

本発明によるプロセッサシステム、アクセス処理方法のいずれにおいても、資源ノードは複数のプロセッサを有するプロセッサノードであることが好ましい。この場合、プロセッサノードは、資源保持部、資源情報保持部として主記憶部、ディレクトリをそれぞれ備えるとともに、複数のプロセッサと主記憶部及びディレクトリに接続されたメモリコントローラを備える。また、制御ノードは、複数の制御デバイスとして複数の入出力デバイスを含む入出力ノードであり、制御コントローラは入出力コントローラである。入出力デバイスは要求としてライトメ

ッセージを発行し、これを受けた入出力コントローラは書き込み要求メッセージをプロセッサノードに向けて出力する。

図面の簡単な説明

図 1 は、従来のマルチプロセッサシステムの構成を示す図である。

図 2 は、従来技術 1 の動作を示すタイミングチャート図である。

図 3 は、従来技術 2 の動作を示すタイミングチャート図である。

図 4 は、本発明のマルチプロセッサシステムの構成を示す図である。

図 5 は、本発明のマルチプロセッサシステムにおける入出力コントローラが、第 I 書き込み許可メッセージを受けた場合の動作を示すフローチャート図である。

図 6 は、本発明のマルチプロセッサシステムの動作として、ライトメッセージが連続して発行された場合の動作を示すタイミングチャート図である。

図 7 は、本発明のマルチプロセッサシステムにおける入出力コントローラの構成を示す図である。

図 8 A は、本発明のマルチプロセッサシステムにおける入出力コントローラの動作例を説明するための図である。

図 8 B は、本発明のマルチプロセッサシステムにおける入出力コントローラの動作例を説明するための図である。

図 8 C は、本発明のマルチプロセッサシステムにおける入出力コントローラの動作例を説明するための図である。

図 8 D は、本発明のマルチプロセッサシステムにおける入出力コントローラの動作例を説明するための図である。

図 8 E は、本発明のマルチプロセッサシステムにおける入出力コントローラの動作例を説明するための図である。

図 8 F は、本発明のマルチプロセッサシステムにおける入出力コントローラの動作例を説明するための図である。

図 8 G は、本発明のマルチプロセッサシステムにおける入出力コントローラの動作例を説明するための図である。

図 8 H は、本発明のマルチプロセッサシステムにおける入出力コントローラの

動作例を説明するための図である。

図8 I は、本発明のマルチプロセッサシステムにおける入出力コントローラの動作例を説明するための図である。

図9 は、本発明のマルチプロセッサシステムの他の構成を示す図である。

図10 は、本発明の実施例1に係るマルチプロセッサシステムの動作として、ディレクトリの状態がUncachedであった場合の動作を示すタイミングチャート図である。

図11 は、本発明の実施例1に係るマルチプロセッサシステムの動作として、ディレクトリの状態がCleanであった場合の動作を示すタイミングチャート図である。

図12 は、本発明の実施例1に係るマルチプロセッサシステムの動作として、ディレクトリの状態がDirtyであった場合の動作を示すタイミングチャート図である。

図13 は、本発明の実施例1に係るマルチプロセッサシステムの動作として、ディレクトリの状態がリクエストロック状態あるいはライトロック状態であった場合の動作を示すタイミングチャート図である。

図14 は、本発明の実施例2に係るマルチプロセッサシステムにおける入出力コントローラの構成を示す図である。

発明を実施するための最良の形態

以下に、本発明のマルチプロセッサシステムについて詳細に説明する。

図4にマルチプロセッサシステムの構成を示す。本発明のマルチプロセッサシステムは、複数のプロセッサノード（資源ノード） $1-1$ 、 \dots 、 $1-i$ 、 \dots 、 $1-m$ （ i 、 m は2以上の整数）と、複数の入出力ノード（制御ノード） $3-1$ 、 \dots 、 $3-j$ 、 \dots 、 $3-n$ （ j 、 n は2以上の整数）とを具備している。複数のプロセッサノード $1-1 \sim 1-m$ と、複数の入出力ノード $3-1 \sim 3-n$ とは、ネットワーク2に接続され、外部からのクロックに応じて動作する。各プロセッサノードは同じ構成を有している。例えば、プロセッサノード $1-1$ は、複数のプロセッサ $10-1-1$ 、 $10-1-2$ と、ディレクトリ（資源情報保持部） 20

ー１と、主記憶部（資源保持部）３０－１と、メモリコントローラ４０－１とを含む。メモリコントローラ４０－１は、プロセッサ１０－１－１、１０－１－２とディレトリ２０－１と主記憶部３０－１とに接続されている。主記憶部３０－１には、複数のデータが格納されている。複数のデータの各々は、その内容を表す値を含んでいる。各入出力ノードも同じ構成を有している。例えば、入出力ノード３－１は、入出力コントローラ（制御コントローラ）５０－１と、外部からの命令によりメッセージを発行する複数の入出力デバイス（制御デバイス）６０－１－１、６０－１－２とを含む。

メッセージは、コマンドの種類を表すコマンド種別と、アドレスとを含む。例えば、メッセージがライトメッセージである場合、コマンド種別はライトを表す。ネットワーク２はメッセージの配送を行い、２点間のメッセージの順序を保障する。なお、以下では、便宜上、図４に示されていない参照番号を使用して説明を行うことがある。例えば、プロセッサノード１－２は図示されていないが、主記憶部３０－２、メモリコントローラ４０－２を含む。

ディレトリ２０－１は、主記憶部３０－１に格納されている各データの一貫性制御に関する情報を、例えば１２８バイトのブロック単位で保持している。他のプロセッサノードにおける主記憶部も同様である。各ブロックの情報としては、一貫性制御に関する状態情報を含む。この状態情報には、アクセス要求を受け付けることができるフリー状態情報と、他のアクセス要求を受け付けることができないライトロック状態情報が含まれる。

一貫性制御について簡単に説明する。マルチプロセッサシステムでは、複数のプロセッサが存在する。また、複数のプロセッサがそれぞれキャッシュを具備し、データのコピーを保持する。そのため、ひとつのデータについて、メモリの値と、コピーを取った複数のキャッシュ値とを一致させる処理が必要になる。このような一致させる処理としては、コピーを無効化する処理が例示される。このように、データの値を一致させる、即ち、データの一貫性を取ることを一貫性制御（一貫性処理）と呼ぶ。データの一貫性制御に関する情報については、後述の実施例にて説明する。

本発明は、入出力コントローラ５０－ｊとメモリコントローラ４０－ｉの間で

の一連のメッセージを工夫することで、ホームを異にするライトメッセージ（要求）の連続発行を可能とするものである。なお、参照番号50-jは入出力コントローラ50-1～50-m（図示せず）を代表して説明するために用いており、参照番号40-iもメモリコントローラ40-1～40-m（図示せず）を代表して説明するために用いている。

ここで、入出力ノード3-1～3-nのうちの第1入出力ノードが入出力ノード3-1であり、その入出力デバイス60-1-1、60-1-2によって1番目からM番目（Mは2以上の整数）までのM個のデータに対するM個のライトメッセージが発行されたものとする。この場合、複数の入出力コントローラのうちの第1入出力コントローラ、つまり入出力ノード3-1の入出力コントローラ50-1がM個のライトメッセージを受け取ると、入出力コントローラ50-1はM個のそれぞれのデータに対するM個のライトトランザクションを開始する。

M個のデータのうちの第Iデータ（I=1、2、…、M）は、プロセッサノード1-1～1-mのうちの1つのプロセッサノードをホームとするデータである。ここでは、この1つのプロセッサノードをプロセッサノード1-iとする。M個のライトメッセージのうちの第Iライトメッセージは、プロセッサノード1-iの主記憶部30-iに格納された複数のデータのうちの第Iデータの値を第Iライトメッセージで指定される値に更新するための命令である。以降、第Iライトメッセージを例にとり説明する。

第Iライトメッセージを受けた入出力コントローラ50-1は、第Iライトトランザクションを開始して、第I書き込み要求メッセージを、ネットワーク2を介してプロセッサノード1-iに出力する。プロセッサノード1-iのメモリコントローラ40-iは、第I書き込み要求メッセージを受け取ると以下のように動作する。メモリコントローラ40-iは、第Iデータに対して、フリー状態情報に代えて、第Iデータに対する他の書き込み要求メッセージを受け付けることができないライトロック状態情報をプロセッサノード1-iのディレクトリ20-iに格納する。メモリコントローラ40-iはまた、第I書き込み要求メッセージに対して第I書き込み許可メッセージを入出力ノード3-1にネットワーク2を介して出力する。

入出力ノード3-1の入出力コントローラ50-1は、第I書き込み許可メッセージを受け取ると、図5に示されるような発行処理を実行する。

図5を参照して、入出力コントローラ50-1は、第1書き込み許可メッセージから第I書き込み許可メッセージまでのI個の書き込み許可メッセージを既に受け取っているか否か进行检查する(ステップS1)。入出力コントローラ50-1は、上記のI個の書き込み許可メッセージを未だ受け取っていない場合(ステップS1のNO)、発行処理を終了し、次の書き込み許可メッセージの到着を待つ。入出力コントローラ50-1は、上記のI個の書き込み許可メッセージを既に受け取っている場合(ステップS1のYES)、第Iライトメッセージで指定される値を含む第I更新メッセージをプロセッサノード1-iにネットワーク2を介して出力する(ステップS2)。

次に、入出力コントローラ50-1は、 $I = I + 1$ とし(ステップS3)、続いてIがM以下であるかどうか进行检查する(ステップS4)。入出力コントローラ50-1は、IがM以下、即ち、第Iライトメッセージを受け取って第Iライトトランザクションを開始していれば(ステップS4のYES)、ステップS1に戻る。入出力コントローラ50-1は、IがMより大、即ち、まだ第Iライトメッセージを受け取っていなければ(ステップS4のNO) 処理を終了する。

このように、入出力コントローラ50-1は、ステップS1~S4を繰り返し実行し、現在のライトトランザクションおよび先行するライトトランザクションが全て書き込み許可メッセージを受け取っていれば、更新メッセージを発行する。

プロセッサノード1-iのメモリコントローラ40-iは、第I更新メッセージを受け取ると、第Iデータに対してライトロック状態情報に代えてフリー状態情報をプロセッサノード1-iのディレクトリ20-iに格納する。メモリコントローラ40-iはまた、プロセッサノード1-iの主記憶部30-iに格納された第Iデータの値を第I更新メッセージで指定される値に更新する。

このように、本発明のマルチプロセッサシステムによれば、入出力コントローラ50-jが、異なるプロセッサノード1-iを宛先とする入出力デバイス60-j-1、60-j-2からの複数のライトメッセージを連続して処理することができる。

図6を参照しながら、順序制約のあるライトメッセージが連続して発行された場合の動作を説明する。図6は、入出力デバイス60-1-1が、M個(M=3)のデータとしてデータA、B、Cに対するライトメッセージをそれぞれステップ1、2、3で発行した場合の動作を示している。ここで、データAおよびデータBはプロセッサノード1-1をホームとし、データCはプロセッサノード1-2をホームとするデータであるとする。また、1ステップは1クロックに対応する。

入出力コントローラ50-1は、ステップ2にて、第1ライトメッセージとしてライトAメッセージを受け取る。この場合、入出力コントローラ50-1は、ライトAトランザクションを開始し、ステップ3にて、ホームのメモリコントローラ40-1に宛てて第1書き込み要求メッセージとして書き込み要求Aメッセージをネットワーク2に出力する。

メモリコントローラ40-1は、ステップ4にて、入出力コントローラ50-1からの書き込み要求Aメッセージを受け取ると、ディレクトリ20-1が保持するデータAの状態情報をフリー状態情報からライトロック状態情報に更新する。続いて、メモリコントローラ40-1は、ステップ5にて、入出力コントローラ50-1に宛てて、第1書き込み許可メッセージとして書き込み許可Aメッセージをネットワーク2に出力する。

入出力コントローラ50-1は、ステップ6にて、書き込み許可Aメッセージを受け取る。このとき、先行するライトメッセージが存在しないので、入出力コントローラ50-1は、ステップ7にて、第1ライトメッセージで指定される値を含む第1更新メッセージとして、更新Aメッセージをメモリコントローラ40-1に宛ててネットワーク2に出力する。

メモリコントローラ40-1は、ステップ8にて、入出力コントローラ50-1からの更新Aメッセージを受け取る。この場合、メモリコントローラ40-1は、ディレクトリ20-1が保持するデータAの状態情報をライトロック状態情報からフリー状態情報に更新し、主記憶部30-1のデータの値を更新Aメッセージで指定される値に更新する。かくして、入出力コントローラ50-1からのライトAメッセージであるデータAが格納される。

入出力コントローラ50-1はまた、ステップ3にて、第2ライトメッセージ

としてライトBメッセージを受け取る。この場合、入出力コントローラ50-1は、ライトBトランザクションを開始し、ステップ4にて、ホームのメモリコントローラ40-1に宛てて、第2書き込み要求メッセージとして書き込み要求Bメッセージをネットワーク2に出力する。

メモリコントローラ40-1は、ステップ5にて、入出力コントローラ50-1からの書き込み要求Bメッセージを受け取ると、ディレクトリ20-1が保持するデータBの状態情報をフリー状態情報からライトロック状態情報に更新する。メモリコントローラ40-1はまた、ステップ6にて、入出力コントローラ50-1に宛てて、第2書き込み許可メッセージとして書き込み許可Bメッセージをネットワーク2に出力する。

入出力コントローラ50-1は、ステップ7にて、書き込み許可Bメッセージを受け取る。この場合、入出力コントローラ50-1は、先行するライトAトランザクションの進捗を検査する。入出力コントローラ50-1は、ステップ7より前のステップ6で既に第1書き込み許可メッセージである書き込み許可Aメッセージを受け取っている。よって、入出力コントローラ50-1は、ステップ8にて、第2ライトメッセージで指定される値を含む第2更新メッセージとして、更新Bメッセージをメモリコントローラ40-1に宛ててネットワーク2に出力する。

メモリコントローラ40-1は、ステップ9にて、入出力コントローラ50-1からの更新Bメッセージを受け取る。この場合、メモリコントローラ40-1は、ディレクトリ20-1が保持するデータBの状態情報をライトロック状態情報からフリー状態情報に更新し、主記憶部30-1のデータの値を更新Bメッセージで指定される値に更新する。かくして、入出力コントローラ50-1からのライトBメッセージであるデータBが格納される。

入出力コントローラ50-1は更に、ステップ4にて、第3ライトメッセージとしてライトCメッセージを受け取る。この場合、入出力コントローラ50-1は、ライトCトランザクションを開始し、ステップ5にて、ホームのメモリコントローラ40-2（図示せず）に宛てて、第3書き込み要求メッセージとして書き込み要求Cメッセージをネットワーク2に出力する。

メモリコントローラ40-2は、ステップ6にて、入出力コントローラ50-1からの書き込み要求Cメッセージを受け取ると、ディレクトリ20-2が保持するデータCの状態情報をフリー状態情報からライトロック状態情報に更新する。メモリコントローラ40-1はまた、ステップ7にて、入出力コントローラ50-1に宛てて、第3書き込み許可メッセージとして書き込み許可Cメッセージをネットワーク2に出力する。

入出力コントローラ50-1は、ステップ8にて、書き込み許可Cメッセージを受け取る。この場合、入出力コントローラ50-1は、先行するライトAトランザクションおよびライトBトランザクションの進捗を検査する。入出力コントローラ50-1は、ステップ8より前のステップ6で既に書き込み許可Aメッセージを受け取り、ステップ8より前のステップ7で既に書き込み許可Bメッセージを受け取っている。よって、入出力コントローラ50-1は、ステップ9にて、第3ライトメッセージで指定される値を含む第3更新メッセージとして、更新Cメッセージをメモリコントローラ40-2に宛ててネットワーク2に出力する。

メモリコントローラ40-2は、ステップ10にて、入出力コントローラ50-1からの更新Cメッセージを受け取る。この場合、メモリコントローラ40-2は、ディレクトリ20-2が保持するデータCの状態情報をライトロック状態情報からフリー状態情報に更新し、主記憶部30-2のデータの値を更新Cメッセージで指定される値に更新する。かくして、入出力コントローラ50-1からのライトCメッセージであるデータCが格納される。

入出力コントローラの構成を入出力コントローラ50-jを例示して説明する。入出力コントローラ50-jは、図7に示されるように、セクタ71-jと、メッセージ格納キュー72-jと、ライトポインタ73-jと、リードポインタA74-jと、リードポインタB75-jとを含む。入出力コントローラ50-jは、複数の入出力デバイス60-j-1、60-j-2からのメッセージをセクタ71-jで調停してメッセージ格納キュー72-jに書き込む。メッセージ格納キュー72-jにおける書き込みの制御はライトポインタ73-jを用いて行われる一方、読み出しの制御はリードポインタA74-jとリードポインタB75-jの二つを用いて行われる。また、入出力コントローラ50-jは、ネ

ットワーク 2 から送られてくるメッセージに応じて許可フラグ 76-j をメッセージ格納キュー 72-j に設定 (格納) し、メッセージ格納キュー 72-1 の制御にこの許可フラグ 76-j も用いる。以上の説明は、例えば入出力コントローラ 50-1 の場合もまったく同じである。つまり、図示していないが、入出力コントローラ 50-1 は、セクタ 71-1 と、メッセージ格納キュー 72-1 と、ライトポインタ 73-1 と、リードポインタ A 74-1 と、リードポインタ B 75-1 とを含む。

図 8A~図 8I は、図 6 に示した動作フローでの入出力コントローラ 50-1 のメッセージ格納キュー 72-1、ライトポインタ 73-1、リードポインタ A 74-1、リードポインタ B 75-1、許可フラグ 76-1 の値の遷移を示す。

図 8A に示されるように、初期状態のステップ ST1 では、ライトポインタ 73-1、リードポインタ A 74-1、リードポインタ B 75-1 は全て「0」を指している。

図 8B に示されるように、入出力コントローラ 50-1 は、ステップ ST2 にて、入出力デバイス 60-1-1 からライト A メッセージを受け取ると、ライトポインタ 73-1 が指すメッセージ格納キュー 72-1 のエン트리「0」にライト A メッセージを書き込む。入出力コントローラ 50-1 はまた、許可フラグ 76-1 のエン트리「0」の値を「0」に設定し、ライトポインタ 73-1 を「1」に更新する。

図 8C に示されるように、入出力コントローラ 50-1 は、ステップ ST3 にて、入出力デバイス 60-1-1 からライト B メッセージを受け取ると、ライトポインタ 73-1 が指すメッセージ格納キュー 72-1 のエン트리「1」にライト B メッセージを書き込む。入出力コントローラ 50-1 はまた、許可フラグ 76-1 のエン트리「1」の値を「0」に設定し、ライトポインタ 73-1 を「2」に更新する。

ここで、ステップ ST2 においてメッセージ格納キュー 72-1 に有効なエン트리が存在する。このため、図 8C に示されるように、入出力コントローラ 50-1 は、ステップ ST3 にて、リードポインタ A 74-1 が示すエン트리「0」の情報により、書き込み要求 A メッセージをネットワーク 2 に出力し、リードポ

インタA 74-1の値を「1」に更新する。

入出力コントローラ50-1は、ステップST4にて、入出力デバイス60-1-1からライトCメッセージを受け取る。この場合、図8Dに示されるように、入出力コントローラ50-1は、ライトポインタ73-1が指すメッセージ格納キュー72-1のエントリ「2」にライトCメッセージを書き込む。入出力コントローラ50-1はまた、許可フラグ76-1のエントリ「2」の値を「0」に設定し、ライトポインタ73-1を「3」に更新する。

また、ステップST3においてメッセージ格納キュー72-1に有効なエントリが存在する。このため、図8Dに示されるように、入出力コントローラ50-1は、ステップST4にて、リードポインタA 74-1が示すエントリ「1」の情報により、書き込み要求Bメッセージをネットワーク2に出力し、リードポインタA 74-1の値を「1」から「2」に更新する。

ステップST4においてメッセージ格納キュー72-1に有効なエントリが存在する。このため、図8Eに示されるように、入出力コントローラ50-1は、ステップST5にて、リードポインタA 74-1が示すエントリ「2」の情報により、書き込み要求Cメッセージをネットワーク2に出力し、リードポインタA 74-1の値を「2」から「3」に更新する。

図8Fに示されるように、入出力コントローラ50-1は、ステップST6にて、書き込み許可Aメッセージを受け取ると、許可フラグ76-1のデータAに該当するエントリ「0」の値を「0」から「1」に更新する。入出力コントローラ50-1はまた、リードポインタB 75-1が指すエントリと、書き込み許可メッセージを受け取ったデータAが格納されているエントリとが一致しているかどうかを検査する。これらが一致するということは、先行するライトトランザクションが存在しない、あるいは既に書き込み許可メッセージを受け取って更新メッセージを発行し完了していることを示している。ここではエントリ「0」で一致するので、入出力コントローラ50-1は更新メッセージの発行処理を行い、次の書き込み許可メッセージの到着を待つ。

入出力コントローラ50-1は、ステップST6にてリードポインタB 75-1が指すメッセージ格納キュー72-1のエントリ「0」の値を読み出し、ステ

ップST7にて、ネットワーク2に更新Aメッセージを出力する。この場合、図8Gに示されるように、入出力コントローラ50-1は、リードポインタB75-1の値を「0」から「1」に更新する。

次に、入出力コントローラ50-1は、リードポインタB75-1の値とライトポインタ73-1の値とを比較する。比較の結果、リードポインタB75-1の値が小さく未完了の書き込みが存在することを示す場合、入出力コントローラ50-1は未完了のライトランザクションで既に書き込み許可メッセージを受け取り済みのものが存在するかどうかを検査する。ここでは、リードポインタB75-1の値「1」はライトポインタ73-1の値「3」より小さいので、リードポインタB75-1が指す許可フラグ76-1のエントリ「1」の値が読み出される。許可フラグ76-1のエントリ「1」の値は「0」であり書き込み許可メッセージをまだ受け取っていないことを示すので、入出力コントローラ50-1はエントリ「1」の更新メッセージ発行処理は行わず次の書き込み許可メッセージの到着を待つ。

図8Gに示されるように、入出力コントローラ50-1は、ステップST7にて、許可Bメッセージを受け取ると、許可フラグ76-1のデータBに該当するエントリ「1」の値を「0」から「1」に更新する。入出力コントローラ50-1はまた、リードポインタB75-1が指すエントリと書き込み許可メッセージを受け取ったデータBが格納されているエントリとが一致しているかどうかを検査する。ここではエントリ「1」で一致するので、入出力コントローラ50-1は更新メッセージの発行処理を行う。

入出力コントローラ50-1は、ステップST7にてリードポインタB75-1が指すメッセージ格納キュー72-1のエントリ「1」の値を読み出し、ステップST8にて、ネットワーク2にライトBメッセージを出力する。この場合、図8Hに示されるように、入出力コントローラ50-1は、リードポインタB75-1の値を「1」から「2」に更新する。

次に、入出力コントローラ50-1はリードポインタB75-1の値「2」とライトポインタ73-1の値「3」を比較する。比較の結果、リードポインタB75-1の値が小さいので、入出力コントローラ50-1はリードポインタB7

5-1が指す許可フラグ76-1のエントリ「2」の値を読み出す。エントリ「2」の値は「0」であり書き込み許可メッセージをまだ受け取っていないことを示すので、入出力コントローラ50-1はエントリ「2」の更新メッセージ発行処理は行わず次の書き込み許可メッセージの到着を待つ。

図8Hに示されるように、入出力コントローラ50-1は、ステップST8にて、書き込み許可Cメッセージを受け取ると、許可フラグ76-1のデータCに該当するエントリ「2」の値を「0」から「1」に更新する。入出力コントローラ50-1はまた、リードポインタB75-1が指すエントリと書き込み許可メッセージを受け取ったデータCが格納されているエントリとが一致しているかどうかを検査する。ここではエントリ「2」で一致するので、入出力コントローラ50-1は更新メッセージの発行処理を行う。

入出力コントローラ50-1は、ステップST8にてリードポインタB75-1が指すメッセージ格納キュー72-1のエントリ「2」の値を読み出し、ステップST9にて、ネットワーク2に更新Cメッセージを出力する。この場合、図8Iに示されるように、入出力コントローラ50-1は、リードポインタB75-1の値を「2」から「3」に更新する。

次に、入出力コントローラ50-1はリードポインタB75-1の値「3」とライトポインタ73-1の値「3」とを比較する。比較の結果、値が一致し未完了のライトトランザクションが存在しないことを示すので、入出力コントローラ50-1は処理を終了する。

上記のように動作することで、ホームを同じにするライトトランザクションの順序も、異なるホームへのライトトランザクションの順序も保障することができる。

ライトBトランザクションとライトAトランザクションの順序関係は次のような理由で保障される。更新Bメッセージを発行する時点で、既に書き込み許可Aメッセージを受け取り済みであり、ホームのディレクトリ20-1の状態情報が、ライトロック状態に遷移していることが保障される。ライトロック状態に遷移していれば、プロセッサや入出力デバイスは更新Aメッセージを受け取ってデータAの値が更新されフリー状態に遷移した後の更新された値しか読み出すことがで

きない。よって、データBの更新された値が読み出せる時点で、データAの更新された値しか読み出すことができないので、順序が保障されたことになる。

ライトCトランザクションとライトBトランザクションの順序関係も同じ理由で保障される。更新Cメッセージを発行する時点で、既に書き込み許可Bメッセージを受け取り済みであり、ホームのディレクトリ20-1の状態情報が、ライトロック状態に遷移していることが保障される。ライトロック状態に遷移していれば、プロセッサや入出力デバイスは更新Bメッセージを受け取ってデータBの値が更新されフリー状態に遷移した後の更新された値しか読み出すことができない。よって、データCの更新された値が読み出せる時点で、データBの更新された値しか読み出すことができないので、順序が保障されたことになる。

また、上記処理は10ステップで完了しており、従来技術1、2と比べて入出力デバイスからのライトメッセージの処理性能を向上させることができる。

上記の例では入出力コントローラが入出力ノード内に設けられているが、入出力コントローラは入出力ノード外でネットワーク2に接続されてもよい。図9はその例を示す。入出力コントローラ52-1、…、52-j、…、52-nがそれぞれ、入出力ノード3-1、…、3-j、…、3-nの外でネットワーク2に接続されている。その他の構成は図4に示された例と同じである。ここで、入出力ノード3-jについて言えば、図7で説明した入出力セクタ71-jに対応する入出力セクタ51-jを備えることが好ましい。入出力セクタ51-jは、複数の入出力デバイス60-j-1、60-j-2が出力するメッセージを調停してネットワーク2に出力する。入出力コントローラ52-jは、ネットワーク2に出力されたメッセージを受けて、図4で説明した入出力コントローラ50-jと同様の処理を行う構成でも良い。これは図9に示されたすべての入出力コントローラ、入出力ノードに当てはまる。

また、図9に示した入出力コントローラ52-jは、ネットワーク2に接続されるのではなく、入出力コントローラ52-i（図示せず）としてプロセッサノード1-i内に備えられても良い。この場合、入出力ノード3-jには、上記の入出力セクタ51-jが設けられていることが好ましい。これは、他のコントローラについても同様である。

また、図4や図9に示す構成で、 m と n とが等しく、プロセッサノード $1-i$ と入出力ノード $3-j$ とで一つのノード（図示せず）を構成してもよい。この場合、そのノードには、ネットワーク2が接続され、上記の入出力セクタ $51-j$ が備えられることが好ましい。勿論、これも他のプロセッサノードと入出力ノードにも当てはまる。

以上の説明で明らかなように、本発明のマルチプロセッサシステムによれば、入出力コントローラ $50-j$ が、異なるプロセッサノード $1-i$ を宛先とする入出力デバイス $60-j-1$ 、 $60-j-2$ からの複数のライトメッセージを連続して処理することができる。また、入出力コントローラ $50-j$ が複数のライトメッセージを連続して処理するため、入出力コントローラ $50-j$ がライトメッセージの処理を行う場合に要する時間を従来のそれよりも短くすることができる。前述したように、参照番号 $50-j$ 、 $1-i$ 、 $60-j-1$ 、 $60-j-2$ は代表例としてあげているにすぎない。

以下では、入出力コントローラ $50-j$ とメモリコントローラ $40-i$ 間のやりとりをより詳細に説明する。

[第1実施例]

上述のように、ディレトリ $20-i$ は、主記憶部 $30-i$ に格納されているデータの一貫性制御に関する情報を、例えば128バイトのブロック単位で保持している。データの一貫性制御に関する情報は、主記憶部 $30-i$ の各ブロックの状態情報（データの一貫性制御に関する状態情報）と、マップ情報とを含む。

ブロックの状態情報（データの一貫性制御に関する状態情報）は、上述のように、他のアクセス要求を受け付けることができるフリー状態情報と、他のアクセス要求を受け付けることができないライトロック状態情報とを含む。

フリー状態情報は、例えば、Uncached、Clean、Dirty（以降U、C、Dとも略す）の3つの状態情報からなる。

Uncached状態情報は、複数のプロセッサノード $1-1\sim 1-m$ のうちのどのプロセッサノードもデータをキャッシングしていないことを示す。Clean状態情報は、複数のプロセッサノード $1-1\sim 1-m$ のうちの少なくとも1つのプロセッサノードがデータをキャッシングしていることを示す。Dirty

状態情報は、複数のプロセッサノード $1-1 \sim 1-m$ のうちのある1つのプロセッサノードがデータをキャッシングし、最新のデータはその1つのプロセッサノードにのみ存在することを示す。

また、ブロックの状態情報（データの一貫性制御に関する状態情報）は、他のアクセス要求を受け付けることができない状態情報として、リクエストされたメッセージのみを受け付けることができるリクエストロック状態情報を含むものとする。

本発明において、ライトロック状態情報は、リクエストロック状態情報と同一にしてもよいし異なる状態情報としても良い。以降の説明では二つの異なる状態情報（以降、リクエストロック状態情報をR、ライトロック状態情報をWとも略す）として存在する場合を例にとり説明する。

また、マップ情報は、各プロセッサノード $1-i$ がブロックのデータをキャッシングしているかどうかを示す情報である。プロセッサノード数分のビットを用いて、マップ情報について説明する。図4の構成例として m を3とする。即ち、プロセッサノード $1-1$ 、 $1-2$ 、 $1-3$ が存在するので、マップ情報を3ビットで表現するものとする。

例えば“000”はどのプロセッサノード $1-1$ 、 $1-2$ 、 $1-3$ もデータをキャッシングしていないことを示す。“001”はプロセッサノード $1-1$ がデータをキャッシングしていることを示す。“010”はプロセッサノード $1-2$ がデータをキャッシングしていることを示す。“100”はプロセッサノード $1-3$ がデータをキャッシングしていることを示す。同様に、“110”はプロセッサノード $1-2$ とプロセッサノード $1-3$ がデータをキャッシングしていることを示す。

図10を参照しながら、本発明の第1実施例に係るマルチプロセッサシステムの第1の動作例を説明する。第1の動作例は、メモリコントローラ40-1が書き込み要求Aメッセージを受けたときに、ディレクトリ20-1の該当するブロックの状態情報がUncached状態情報であった場合の動作である。図10は、入出力デバイス60-1-1が、データAに対するライトメッセージをステップ1で発行した場合の動作を示している。ここで、データAはプロセッサノード $1-1$ をホームとするデータであるとする。また、1ステップは1クロックに

対応する。

入出力コントローラ50-1は、ステップ2にて、ライトAメッセージを受け取る。この場合、入出力コントローラ50-1は、ステップ3にて、ホームのメモリコントローラ40-1に宛ててネットワーク2に書き込み要求Aメッセージを出力する。

メモリコントローラ40-1は、ステップ4にて、入出力コントローラ50-1からの書き込み要求Aメッセージを受け取り、ディレクトリ20-1が保持するデータAの状態情報をフリー状態情報からライトロック状態情報に更新する。即ち、ディレクトリ20-1の該ブロックの値を“U、000”から“W、000”に更新する。ここで、“U、000”は状態情報がU (Uncached)であることを示し、マップ情報が“000”であることを示す。また、メモリコントローラ40-1は、ステップ5にて、入出力コントローラ50-1に宛ててネットワーク2に書き込み許可Aメッセージを出力する。

入出力コントローラ50-1は、ステップ6にて、書き込み許可Aメッセージを受け取る。次に、ステップ7にて、更新Aメッセージをメモリコントローラ40-1に宛ててネットワーク2に出力する。

メモリコントローラ40-1は、ステップ8にて、入出力コントローラ50-1からの更新Aメッセージを受け取り、主記憶部30-1のデータの値を更新Aメッセージで指定される値に更新する。そして、ディレクトリ20-1が保持するデータAの状態情報をライトロック状態情報からフリー状態情報に更新する。即ち、ディレクトリ20-1の該当するブロックの値を“W、000”から“U、000”に更新する。

図11を参照しながら、本発明の第1実施例に係るマルチプロセッサシステムの第2の動作例を説明する。第2の動作例は、メモリコントローラ40-1が書き込み要求Aメッセージを受けたときに、ディレクトリ20-1の該当するブロックの状態情報がClean状態情報で、プロセッサノード1-2とプロセッサノード1-3がデータをキャッシングしている場合の動作である。図11は、入出力デバイス60-1-1が、データAに対するライトメッセージをステップ1で発行した場合の動作を示している。ここで、データAはプロセッサノード1-

1をホームとするデータであるとする。また、1ステップは1クロックに対応する。

入出力コントローラ50-1は、ステップ2にて、ライトAメッセージを受け取る。この場合、入出力コントローラ50-1は、ステップ3にて、ホームのメモリコントローラ40-1に宛てて書き込み要求Aメッセージをネットワーク2に出力する。

メモリコントローラ40-1は、ステップ4にて、入出力コントローラ50-1からの書き込み要求Aメッセージを受け取り、ディレクトリ20-1が保持するデータAの状態情報をフリー状態情報からライトロック状態情報に更新する。即ち、ディレクトリ20-1の該当するブロックの値を“C、110”から“W、000”に更新する。ここで、“C、110”は状態情報がC (Clean)であることを示し、マップ情報が“110”であることを示す。また、メモリコントローラ40-1は、ステップ5にて、入出力コントローラ50-1に宛ててネットワーク2に応答Aメッセージを出力し、メモリコントローラ40-2とメモリコントローラ40-3に宛てて無効化Aメッセージを出力する。この場合、応答Aメッセージには、キャッシングしているプロセッサノードの数(この例では2)が付加される。

入出力コントローラ50-1は、ステップ6にて、応答Aメッセージを受け取る。また、メモリコントローラ40-2、メモリコントローラ40-3は、ステップ6にて、それぞれ無効化Aメッセージを受け取り、それぞれプロセッサノード1-2、プロセッサノード1-3でキャッシングしているデータAを無効化する。続いて、メモリコントローラ40-2、メモリコントローラ40-3は、ステップ7にて、それぞれ入出力コントローラ50-1に宛てて無効化完了Aメッセージをネットワーク2に出力する。

入出力コントローラ50-1は、ステップ8にて、応答Aメッセージに付加されている数の無効化完了Aメッセージを受け取った時点で、書き込み許可Aメッセージを受け取ったと解釈(認識)する。入出力コントローラ50-1は、ステップ9にて、更新Aメッセージをメモリコントローラ40-1に宛ててネットワーク2に出力する。

メモリコントローラ 40-1 は、ステップ 10 にて、入出力コントローラ 50-1 からの更新 A メッセージを受け取り、主記憶部 30-1 のデータの値を更新 A メッセージで指定される値に更新する。続いて、メモリコントローラ 40-1 は、ディレクトリ 20-1 が保持するデータ A の状態情報をライトロック状態情報からフリー状態情報に更新する。即ち、ディレクトリ 20-1 の該当するブロックの値を “W、000” から “U、000” に更新する。

図 12 を参照しながら、本発明の第 1 実施例に係るマルチプロセッサシステムの第 3 の動作を説明する。第 3 の動作例は、メモリコントローラ 40-1 が書き込み要求 A メッセージを受けたときに、ディレクトリ 20-1 の該当するブロックの状態情報が Dirty 状態情報で、プロセッサノード 1-2 がデータをキャッシングしている場合の動作である。図 12 は、入出力デバイス 60-1-1 が、データ A に対するライトメッセージをステップ 1 で発行した場合の動作を示している。ここで、データ A はプロセッサノード 1-1 をホームとするデータであるとする。また、1 ステップは 1 クロックに対応する。

入出力コントローラ 50-1 は、ステップ 2 にて、ライト A メッセージを受け取る。この場合、入出力コントローラ 50-1 は、ステップ 3 にて、ホームのメモリコントローラ 40-1 に宛てて書き込み要求 A メッセージをネットワーク 2 に出力する。

メモリコントローラ 40-1 は、ステップ 4 にて、入出力コントローラ 50-1 からの書き込み要求 A メッセージを受け取り、ディレクトリ 20-1 が保持するデータ A の状態情報をフリー状態情報からリクエストロック状態情報に更新する。即ち、メモリコントローラ 40-1 は、ディレクトリ 20-1 の該当するブロックの値を “D、010” から “R、010” に更新する。ここで、“R、010” は状態情報が R (Request Lock) であることを示し、マップ情報が “010” であることを示す。また、メモリコントローラ 40-1 は、ステップ 5 にて、入出力コントローラ 50-1 に宛てて書き戻し要求 A メッセージをネットワーク 2 に出力する。なお、リクエストロック状態情報である “R、010” への更新ではなく、ライトロック状態情報である “W、000” への更新であっても構わない。

メモリコントローラ40-2は、ステップ6にて、メモリコントローラ40-1からの書き戻し要求Aメッセージを受け取り、プロセッサノード1-2でキャッシングしているデータAの書き戻しを行う。続いて、メモリコントローラ40-2は、ステップ7にて、メモリコントローラ40-1に宛てて書き戻しAメッセージをネットワーク2に出力する。

メモリコントローラ40-1は、ステップ8にて、リクエストロック状態情報によりリクエストされたメッセージとして、書き戻しAメッセージをメモリコントローラ40-2から受け取り、ディレクトリ20-1が保持するデータAの状態情報をリクエストロック状態情報からライトロック状態情報に更新する。即ち、ディレクトリ20-1の該当するブロックの値を“R、010”から“W、000”に更新する。また、メモリコントローラ40-1は、ステップ9にて、入出力コントローラ50-1に宛てて書き込み許可Aメッセージをネットワーク2に出力する。

入出力コントローラ50-1は、ステップ10にて、書き込み許可Aメッセージを受け取る。入出力コントローラ50-1は次に、ステップ11にて、更新Aメッセージをメモリコントローラ40-1に宛ててネットワーク2に出力する。

メモリコントローラ40-1は、ステップ11にて、入出力コントローラ50-1からの更新Aメッセージを受け取り、主記憶部30-1のデータの値を更新Aメッセージで指定される値に更新する。続いてメモリコントローラ40-1は、ディレクトリ20-1が保持するデータAの状態情報をライトロック状態情報からフリー状態情報に更新する。即ち、ディレクトリ20-1の該当するブロックの値を“W、000”から“U、000”に更新する。

図13を参照しながら、本発明の第1実施例に係るマルチプロセッサシステムの第3の動作例を説明する。第3の動作例は、メモリコントローラ40-1が書き込み要求Aメッセージを受けたときに、ディレクトリ20-1の該当するブロックの状態情報がリクエストロック状態情報“R”あるいはライトロック状態情報“W”であった場合の動作である。図13は、入出力デバイス60-1-1が、データAに対するライトメッセージをステップ1で発行した場合の動作を示している。ここで、データAはプロセッサノード1-1をホームとするデータである

とする。また、1ステップは1クロックに対応する。

入出力コントローラ50-1は、ステップ2にて、ライトAメッセージを受け取る。この場合、入出力コントローラ50-1は、ステップ3にて、ホームのメモリコントローラ40-1に宛てて書き込み要求Aメッセージをネットワーク2に出力する。

メモリコントローラ40-1は、ステップ4にて、入出力コントローラ50-1からの書き込み要求Aメッセージを受け取る。この場合、ディレクトリ20-1が保持するデータAの状態情報は、リクエストロック状態情報“R”あるいはライトロック状態情報“W”である。このため、メモリコントローラ40-1は、ステップ5にて、入出力コントローラ50-1に宛てて不許可Aメッセージをネットワーク2に出力する。

入出力コントローラ50-1は、ステップ6にて、不許可Aメッセージを受け取り、ステップ7にて、メモリコントローラ40-1に宛てて書き込み要求Aメッセージをネットワーク2に再出力する。ステップ7以降の動作は、これまで図10から図12を参照しながら説明した動作と同じであるので省略する。

以上のように動作することで、ディレクトリのマップ情報に入出力ノード分のビットを加えなくて済む。また、ホームが受けたプロセッサからのアクセス要求を、入出力コントローラ50-jに対して転送せずに済み、メモリコントローラや入出力コントローラの構成が複雑にならずに済む。

[第2実施例]

本発明によるマルチプロセッサシステムの入出力コントローラは図14に示すように構成されても良い。図14を参照して入出力コントローラ50-jに適用した場合について説明する。

図14に示した入出力コントローラ50-jは、図7に示した入出力コントローラ50-jに更に、開放処理ポインタ78-jと開放処理フラグ79-jを付加している。初期状態で、開放処理フラグ79-jの値は“0”である。

基本動作は第1実施例と同様であるので、ここでは異なる動作のみを説明し、同じ動作は省略する。また、最良の実施の形態と同様に、入出力ノード3-1～3-nのうちの第1入出力ノード3-1の入出力デバイス60-1-1、60-

1-2によって1番目からM番目(Mは2以上の整数)までのM個のデータに対するM個のライトメッセージが発行されたものとする。以降、第Iライトメッセージを例にとり説明する。

図4をも参照して、メモリコントローラ40-iは、第Iデータに対して前記ライトロック状態情報がディレクトリ20-iに格納されているときに、第Iデータに対する第I書き込み要求メッセージを入出力コントローラ50-1から受け取った場合、第I書き込み要求メッセージに対して第I開放要求メッセージを入出力コントローラ50-1にネットワーク2を介して出力する。第I開放要求メッセージを受けた入出力コントローラ50-1は、第I書き込み要求メッセージをメモリコントローラ40-iに宛ててネットワーク2に出力する。また、開放処理フラグ79-1(図示せず)の値を“0”から“1”に更新し、開放処理ポインタ78-1の値を、第I番目のライトトランザクションであることを示す情報“I”に設定する。続いて、後述の開放処理を行う。

開放処理フラグ79-1は、入出力コントローラ50-1が後述の開放処理を行っている状態であるかどうかを示す。一方、開放処理ポインタ78-1(図示せず)は、開放処理がどのライトトランザクションによって引き起こされたものであるかを示す。開放処理ポインタ78-1の値は、開放処理を引き起こしたライトトランザクションと他のライトトランザクションとの順序関係(先行するものか後続するものか)を判断するのに用いる。

入出力コントローラ50-1は、開放処理フラグ79-1の値が“1”の間、第Iライトに後続する第Kトランザクションに関して、次の処理を行う。但し、Kは、 $K = I + 1$ 、 $I + 2$ 、 \dots 、Mを満たす整数であり、 $I + 1$ は、 $I < (I + 1) < M$ を満たす整数であり、 $I + 2$ は、 $(I + 1) < (I + 2) < M$ を満たす整数である。

入出力コントローラ50-1は、未だ第K書き込み要求メッセージを発行していない場合、第K書き込み要求メッセージの発行を停止する。

入出力コントローラ50-1は、既に第K書き込み要求メッセージを発行し第K書き込み許可メッセージを受け取っている場合、以下のように動作する。つまり、入出力コントローラ50-1は、許可フラグ76-1(図示せず)の該当す

るエントリの値を“0”に更新し、第K開放メッセージを前記第Kデータのホームであるプロセッサノード1-k ($k=1, 2, \dots, m$) のメモリコントローラ40-k (図示せず) にネットワーク2を介して出力する。

入出力コントローラ50-1は、既に第K書き込み要求メッセージを発行し未だ第K書き込み許可メッセージを受け取っていない場合、第K書き込み許可メッセージを受け取ったときに、許可フラグ76-1の該当するエントリの値を“1”に更新せず、第K開放メッセージの発行を行う。

第K開放メッセージを受け取ったメモリコントローラ40-kは、そのディレクトリ20-k (図示せず) が保持するデータの状態情報をライトロック状態情報からフリー状態情報に更新する。即ち、ディレクトリ20-kの該当するブロックの値を“W、000”から“U、000”に更新する。

既に第K書き込み要求メッセージを発行したがまだ第K許可メッセージを受け取っていない場合、入出力コントローラ50-1は受け取るメッセージによって次のように動作する。入出力コントローラ50-1は第K書き込み許可メッセージを受け取った場合、許可フラグ76-1の該当するエントリの値は“0”のままとし、第K開放メッセージをメモリコントローラ40-kに発行する。第K不許可メッセージあるいは第K開放要求メッセージを受け取った場合はなにもしない。

また、入出力コントローラ50-1は、開放処理フラグ79-1の値が“1”の間、第Iライトランザクションに先行する第1～第(I-1)ライトランザクションに関しては、第1実施例と同じように動作する。ただし、先行する第Lライトランザクションで開放要求メッセージを受け取った場合、入出力コントローラ50-1は第L書き込み要求メッセージをメモリコントローラ40-1に宛ててネットワーク2に出力する。なお、Lは、 $1 \leq L \leq (I-1)$ を満たす整数である。入出力コントローラ50-1はまた、開放処理ポインタ78-1の値を第L番目のライトランザクションであることを示す情報に設定する。そして、第Lライトを基準に、先行するライトランザクションであるか、後続するライトランザクションであるかを判断し、上記開放処理を行う。

入出力コントローラ50-1は開放処理ポインタ78-1の値が示すライトト

ランザクション（ここでは第Iライトランザクションとする）に関して第I書き込み許可メッセージを受け取ると、第I更新メッセージをメモリコントローラ40-iに発行する。入出力コントローラ50-1はまた、開放処理フラグ79-1の値を“0”に更新し、開放処理を終了する。

以上のような動作により、例えば入出力デバイス60-1-1がライトAメッセージ、ライトCメッセージの順に発行し、別の入出力デバイス60-2-2がライトCメッセージ、ライトAメッセージの順に発行した場合のデッドロックの危険性を回避することができる。

〔第3実施例〕

本発明のマルチプロセッサシステムは、以下に示すような動作を実行することもできる。

ディレクトリ（例えば20-i）に格納される情報として、前述した状態情報に加えて、さらにある特定のメッセージの処理のみを受け付ける割り込み可ライトロック状態情報（以降Wiと記す）が存在する。また、割り込み可ライトロック状態時にどの入出力ノード（例えば3-j）が発行したメッセージによって遷移したかを示すフィールドが加わる。以降例えば“Wi、000、2”とした場合、入出力ノード3-2が発行したメッセージによって割り込み可ライトロック状態に遷移したことを示す。

基本動作は第1実施例と同様であるので、ここでは異なる動作のみを説明し、同じ動作は省略する。また、最良の実施の形態と同様に、入出力ノード3-1～3-nのうちの第1入出力ノード3-1の入出力デバイス60-1-1、60-1-2によって1番目からM番目までのM個のデータに対するM個のライトメッセージが発行されたものとする。以降、第Iライトメッセージを例にとり説明する。

入出力コントローラ50-1は、第Iライトメッセージを受けたとき、第1～第(I-1)書き込み許可メッセージまでの(I-1)個の書き込み許可メッセージを既に受け取っているか否か进行检查し、受け取り済みであれば第I強書き込み要求メッセージを、受け取り済みでなければ第I弱書き込み要求メッセージを発行する。

メモリコントローラ 40-i は、第 I データに対してフリー状態情報がディレクトリ 20-i に格納されているときに、第 I 強書き込み要求メッセージを入出力コントローラ 50-1 から受け取った場合、以下のように動作する。つまり、メモリコントローラ 40-i は、前記フリー状態情報に代えて、プロセッサや入出力デバイスからの読み出し要求や、前記第 I データに対する他の強書き込み要求メッセージ及び弱書き込み要求メッセージの両方を受け付けることができないライトロック状態情報をディレクトリ 20-i に格納する。メモリコントローラ 40-i はまた、入出力コントローラ 50-1 に宛てて第 I 書き込み許可メッセージをネットワーク 2 に出力する。

メモリコントローラ 40-i は、第 I データに対してフリー状態情報がディレクトリ 20-i に格納されているときに、第 I 弱書き込み要求メッセージを入出力コントローラ 50-1 から受け取った場合、以下のように動作する。つまり、メモリコントローラ 40-i は、前記フリー状態情報に代えて、プロセッサや入出力デバイスからの読み出し要求や、前記第 I データに対する他の弱書き込み要求メッセージを受け付けることができない割り込み可ライトロック状態情報をディレクトリ 20-i に格納する。メモリコントローラ 40-i はまた、入出力コントローラ 50-1 に宛てて第 I 書き込み許可メッセージをネットワーク 2 に出力する。

メモリコントローラ 40-i は、第 I データに対してライトロック状態情報がディレクトリ 20-i に格納されているときに、第 I 強書き込み要求メッセージを入出力コントローラ 50-1 から受け取った場合、入出力コントローラ 50-1 に宛てて第 I 不許可メッセージをネットワーク 2 に出力する。

メモリコントローラ 40-i は、第 I データに対して割り込み可ライトロック状態情報あるいはライトロック状態情報がディレクトリ 20-i に格納されているときに、第 I 弱書き込み要求メッセージを入出力コントローラ 50-1 から受け取った場合、入出力コントローラ 50-1 に宛てて第 I 不許可メッセージをネットワーク 2 に出力する。

入出力コントローラ 50-1 は、第 I 不許可メッセージを受け取ると、第 1 ～ 第 (I-1) 書き込み許可メッセージまでの (I-1) 個の書き込み許可メッセ

ージを既に受け取っているか否かを検査し、受け取り済みであれば第 I 強書き込み要求メッセージを、受け取り済みでなければ第 I 弱書き込み要求メッセージを発行する。

メモリコントローラ 40-i は、第 I データに対して割り込み可ライトロック状態情報がディレクトリ 20-i に格納されているときに、第 I 強書き込み要求メッセージを入出力コントローラ 50-1 から受け取った場合、前記割り込み可ライトロック状態情報に代えて、リクエストロック状態情報をディレクトリ 20-i に格納する。ディレクトリ 20-i に格納された入出力コントローラを特定する情報が例えば入出力コントローラ 50-2 を指す場合、メモリコントローラ 40-i は、入出力コントローラ 50-2 に宛てて第 I 再試行要求メッセージをネットワークに 2 出力する。

第 I 再試行要求メッセージを受け取った入出力コントローラ 50-2 は、以下に示す再試行処理を行う。

まず、入出力コントローラ 50-2 はメモリコントローラ 40-i に宛てて第 I 開放メッセージをネットワーク 2 に出力する。

入出力コントローラ 50-2 は、次に、受け取ったライトメッセージの中で、前記第 I データとアドレスを同じくするライトメッセージで書き込み許可メッセージを受け取り済みのものに関して、まだ更新メッセージを発行していなければ更新メッセージの発行を停止する。入出力コントローラ 50-2 はまた、前記第 I 開放メッセージの発行後に、メモリコントローラ 40-i に宛てて書き込み要求メッセージを発行する。これで再試行処理は終わる。

メモリコントローラ 40-i は、前記第 I 開放メッセージを受け取ると、リクエストロック状態情報に代えてライトロック状態情報をディレクトリ 20-i に格納する。メモリコントローラ 40-i はまた、入出力コントローラ 50-1 に宛てて第 I 書き込み許可メッセージをネットワーク 2 に出力する。

メモリコントローラ 40-i は、第 I データに対してリクエストロック状態情報がディレクトリ 20-i に格納されているときに、第 I データに対する更新メッセージを入出力コントローラ 50-2 から受け取った場合、主記憶部 30-2 のデータの値を更新メッセージで指定される値に更新する。

第3実施例の変形として、次のような変形も可能である。

第I再試行要求メッセージを受け取った入出力コントローラ50-2が行う再試行処理について説明する。この場合、入出力コントローラ50-2は受け取ったライトメッセージの中で、前記第Iデータとアドレスを同じくするライトメッセージで書き込み許可メッセージを受け取り済みのものに関して、まだ更新メッセージを発行していなければ第I開放メッセージを発行する。入出力コントローラ50-2は、その後にメモリコントローラ40-iに宛てて書き込み要求メッセージを発行し処理を終える。入出力コントローラ50-2は、まだ更新メッセージを発行していればなんにもせず処理を終える。

メモリコントローラ40-iは、前記第I開放メッセージを受け取ると、リクエストロック状態情報に代えてライトロック状態情報をディレクトリ20-iに格納する。メモリコントローラ40-iはまた、入出力コントローラ50-1に宛てて第I書き込み許可メッセージをネットワーク2に出力する。

メモリコントローラ40-iは、第Iデータに対してリクエストロック状態情報がディレクトリ20-iに格納されているときに、第Iデータに対する更新メッセージを入出力コントローラ50-2から受け取った場合、主記憶部30-2のデータの値を更新メッセージで指定される値に更新する。メモリコントローラ40-iはまた、リクエストロック状態情報に代えてライトロック状態情報をディレクトリ20-iに格納する。続いて、メモリコントローラ40-iは、入出力コントローラ50-1に宛てて第I書き込み許可メッセージをネットワーク2に出力する。

以上のように構成及び動作により、第2実施例と同じようにデッドロックの危険性を回避することができる。

以上説明してきたように、本発明のマルチプロセッサシステム及びメモリアクセス処理方法によれば、入出力コントローラが、異なるプロセッサノードを宛先とする入出力デバイスからの複数のライトメッセージを連続して処理することができる。また、本発明のマルチプロセッサシステム及びメモリアクセス処理方法によれば、入出力コントローラが複数のライトメッセージを連続して処理するため、入出力コントローラがライトメッセージの処理を行う場合に要する時間を従

来のそれよりも短くすることができる。

以下に、本発明によるマルチプロセッサシステムのいくつかの態様を記載する。

〔第1の態様〕

マルチプロセッサシステムは、ネットワークに接続された複数のプロセッサノードと、ネットワークに接続された複数の入出力ノードと、複数の入出力コントローラとを具備する。

複数のプロセッサノードは、それぞれ複数のプロセッサと、複数のデータを格納する主記憶部と、複数のデータの各々に対してアクセス要求を受け付けることが可能なフリー状態情報が格納されたディレクトリと、複数のプロセッサと主記憶部とディレクトリとに接続されたメモリコントローラとを含む。

複数の入出力ノードは、それぞれライトメッセージを発行する複数の入出力デバイスを含む。

複数の入出力ノードのうちの第1入出力ノードの複数の入出力デバイスによって1番目からM番目（Mは2以上の整数）までのM個のデータに対するM個のライトメッセージが発行されたとき、複数の入出力コントローラのうちの第1入出力コントローラは、M個のそれぞれのデータに対するM個のライトトランザクションを開始する。M個のデータのうちの第Iデータ（Iは、 $I=1, 2, \dots, M$ を満たす整数）は、複数のプロセッサノードのうちの第1プロセッサノードをホームとするデータである。第Iライトメッセージは第1プロセッサノードの主記憶部に格納された複数のデータのうちの第Iデータの値を第Iライトメッセージで指定される値に更新するための命令である。第1入出力コントローラは、第Iライトトランザクションの処理として、第I書き込み要求メッセージを第1プロセッサノードにネットワークを介して出力する。

第1プロセッサノードのメモリコントローラは、第I書き込み要求メッセージを受け取ったとき、第Iデータに対して、フリー状態情報に代えて、第Iデータに対するプロセッサや入出力デバイスからの読み出し要求や他の書き込み要求メッセージを受け付けることができないライトロック状態情報を第1プロセッサノードのディレクトリに格納し、第I書き込み要求メッセージに対して第I書き込み許可メッセージを第1入出力コントローラにネットワークを介して出力する。

第1入出力コントローラは、第I書き込み許可メッセージを受け取ったときに第Iライトトランザクションの更新メッセージ発行処理を行う。第1入出力コントローラはまた、更新メッセージ発行処理において、第1から第I書き込み許可トランザクションまでのI個の書き込み許可メッセージを既に受け取っているかをチェックする。第1入出力コントローラは、I個の書き込み許可メッセージをまだ受け取っていないならば、第I更新メッセージ発行処理を終了する。第1入出力コントローラはまた、I個の書き込み許可メッセージを既に受け取っているとき、第Iライトメッセージで指定される値を含む第I更新メッセージを第1プロセッサノードにネットワークを介して出力して第Iライトトランザクションを完了させる。第1入出力コントローラは更に、 $(I + 1)$ がM以下であれば第 $(I + 1)$ ライトトランザクションの更新メッセージ発行処理を行い、 $(I + 1)$ がMより大きければ第I更新メッセージ発行処理を終了する。

第1プロセッサノードのメモリコントローラは、第I更新メッセージを受け取ったとき、第Iデータに対してライトロック状態情報に代えてフリー状態情報を第1プロセッサノードのディレクトリに格納すると共に、第1プロセッサノードの主記憶部に格納された第Iデータの値を第I更新メッセージで指定される値に更新する。

[第2の態様]

第1の態様によるマルチプロセッサシステムにおいて、複数の入出力コントローラは、それぞれ複数の入出力ノード内に備えられる。複数の入出力ノードの各々の入出力コントローラには、複数の入出力ノードの各々の複数の入出力デバイスによって発行されるM個のライトメッセージを調停するセレクトが備えられる。

[第3の態様]

第1の態様によるマルチプロセッサシステムにおいて、複数の入出力コントローラはネットワークに接続される。複数の入出力ノードの各々には、複数の入出力ノードの各々の複数の入出力デバイスによって発行されるM個のライトメッセージを調停するセレクトが備えられる。

[第4の態様]

第1の態様によるマルチプロセッサシステムにおいて、複数の入出力コントロ

ーラは、それぞれ複数のプロセッサノード内に備えられる。複数の入出力ノードの各々には、複数の入出力ノードの各々の複数の入出力デバイスによって発行されるM個のライトメッセージを調停するセクタが備えられる。

[第5の態様]

第1の態様によるマルチプロセッサシステムにおいて、複数のプロセッサノードと複数の入出力ノードとが、それぞれ複数のノードを構成する。複数のノードの各々には、複数のノードの各々の複数の入出力デバイスによって発行されるM個のライトメッセージを調停するセクタが備えられる。

第1～第5の態様のいずれかによるマルチプロセッサシステムにおいて、第1プロセッサノードのメモリコントローラは、第Iデータに対してライトロック状態情報が第1プロセッサノードのディレクトリに格納されているときに、第Iデータに対する第I書き込み要求メッセージを第1入出力コントローラから受け取った場合、第I書き込み要求メッセージに対して第I開放要求メッセージを第1入出力コントローラにネットワークを介して出力する。

第1入出力コントローラは、第I開放要求メッセージを受けて第I書き込み要求メッセージを第1プロセッサノードのメモリコントローラにネットワークを介して出力すると共に、開放処理を行う。

第1入出力コントローラは、開放処理において、第Kライトトランザクション{Kは、 $K=I+1$ 、 $I+2$ 、 \dots 、Mを満たす整数、 $I+1$ は、 $I < (I+1) < M$ を満たす整数、 $I+2$ は、 $(I+1) < (I+2) < M$ を満たす整数}の進捗を検査する。

第1入出力コントローラは、未だ第K書き込み要求メッセージを発行していない場合、第K書き込み要求メッセージの発行を停止する。

第1入出力コントローラは、既に第K書き込み要求メッセージを発行し第K書き込み許可メッセージを受け取っている場合、第K開放メッセージを第Kデータのホームである第2プロセッサノードにネットワークを介して出力する。

第1入出力コントローラは、既に第K書き込み要求メッセージを発行しまだ第K書き込み許可メッセージを受け取っていない場合は、第K書き込み許可メッセージを受け取った時点で第K開放メッセージの発行を行う。

第2プロセッサノードのメモリコントローラは、第K開放メッセージを受け取ったとき、第Kデータに対してライトロック状態情報に代えてフリー状態情報を第2プロセッサノードのディレクトリに格納する。

[第7の態様]

第1～第5の態様のいずれかによるマルチプロセッサシステムにおいて、ディレクトリに格納される状態情報は、さらに割り込み可ライトロック状態情報とリクエストロック状態情報を含む。

第1入出力コントローラは第I書き込み要求メッセージを発行する際に、第1から第(I-1)書き込み許可メッセージまでの(I-1)個の書き込み許可メッセージを既に受け取っているか否かを検査し、受け取り済みであれば第I強書き込み要求メッセージを、受け取り済みでなければ第I弱書き込み要求メッセージを発行する。

第1プロセッサノードのディレクトリに格納されている第Iデータの状态情報がフリー状態情報であるときに、第1プロセッサノードのメモリコントローラは、第I強書き込み要求メッセージを第1入出力コントローラから受け取った場合、フリー状態情報に代えて、プロセッサや入出力デバイスからの読み出し要求や、第Iデータに対する他の強・弱書き込み要求メッセージ両方を受け付けることができないライトロック状態情報を第1プロセッサノードのディレクトリに格納し、第I書き込み許可メッセージを第1入出力コントローラにネットワークを介して出力する。

第1プロセッサノードのディレクトリに格納されている第Iデータの状态情報がフリー状態情報であるときに、第1プロセッサノードのメモリコントローラは、第I弱書き込み要求メッセージを第1入出力コントローラから受け取った場合、フリー状態情報に代えて、プロセッサや入出力デバイスからの読み出し要求や、第Iデータに対する他の弱書き込み要求メッセージを受け付けることができない割り込み可ライトロック状態情報を第1プロセッサノードのディレクトリに格納し、第I書き込み許可メッセージを第1入出力コントローラにネットワークを介して出力する。

割り込み可ライトロック状態情報は、この状態にした弱書き込み要求メッセー

ジを発行した入出力コントローラを特定する情報を含む。

第1プロセッサノードのディレクトリに格納されている第Iデータの状態情報がライトロック状態情報あるいはリクエストロック状態情報であるときに、第1プロセッサノードのメモリコントローラは、第I強書き込み要求メッセージを第1入出力コントローラから受け取った場合、第I不許可メッセージを第1入出力コントローラにネットワークを介して出力する。

第1プロセッサノードのディレクトリに格納されている第Iデータの状態情報が割り込み可ライトロック状態情報、ライトロック状態情報、あるいはリクエストロック状態情報であるときに、第1プロセッサノードのメモリコントローラは、第I弱書き込み要求メッセージを第1入出力コントローラから受け取った場合、第I不許可メッセージを第1入出力コントローラにネットワークを介して出力する。

第1入出力コントローラは、第I不許可メッセージを受け取ったとき、第1から第(I-1)書き込み許可メッセージまでの(I-1)個の書き込み許可メッセージを既に受け取っているか否かを検査し、受け取り済みであれば第I強書き込み要求メッセージを、受け取り済みでなければ第I弱書き込み要求メッセージを発行する。

第1プロセッサノードのディレクトリに格納されている第Iデータの状態情報が割り込み可ライトロック状態情報であるときに、第1プロセッサノードのメモリコントローラは、第I強書き込み要求メッセージを第1入出力コントローラから受け取った場合、割り込み可ライトロック状態情報に代えて、プロセッサや入出力デバイスからの読み出し要求や、第Iデータに対する他の弱書き込み要求メッセージを受け付けることができないリクエストロック状態情報を第1プロセッサノードのディレクトリに格納し、第I再試行要求メッセージをディレクトリに格納された入出力コントローラを特定する情報が指す第2入出力コントローラに宛ててネットワークを介して出力する。

第I再試行要求メッセージを受け取った第2入出力コントローラは、再試行処理を行う。

[第8の態様]

第7の態様によるマルチプロセッサシステムにおいて、第2入出力コントローラが行う再試行処理は以下の通りである。

はじめに、第I開放メッセージを第1プロセッサノードのメモリコントローラにネットワークを介して出力する。続いて、第Iデータとアドレスを同じくするライトトランザクションで書き込み許可メッセージを受け取り済みのものに関して、まだ更新メッセージを発行していなければ更新メッセージの発行を停止して、第I開放メッセージの発行後に、第1プロセッサノードに宛てて書き込み要求メッセージを発行する。第I開放メッセージを受けた第1プロセッサノードのメモリコントローラは、リクエストロック状態情報に代えてライトロック状態情報を第1プロセッサノードのディレクトリに格納し、第I書き込み許可メッセージを第1入出力コントローラに宛ててネットワークを介して出力する。

〔第9の態様〕

本発明によるメモリアクセス処理方法は以下のマルチプロセッサシステムに適用される。マルチプロセッサシステムは、複数のプロセッサノードと、複数の入出力ノードと、複数の入出力コントローラとを具備する。複数のプロセッサノードの各々は、複数のプロセッサと、複数のデータを格納する主記憶部と、複数のデータの各々に対してアクセス要求を受け付けることが可能なフリー状態情報が格納されたディレクトリと、複数のプロセッサと主記憶部とディレクトリとに接続されたメモリコントローラとを含む。複数の入出力ノードには、それぞれライトメッセージを発行する複数の入出力デバイスが備えられる。

本メモリアクセス処理方法においては、複数の入出力ノードのうちの第1入出力ノードの複数の入出力デバイスによって1番目からM番目(Mは2以上の整数)までのM個のデータに対するM個のライトメッセージが発行されたとき、複数の入出力コントローラのうちの第1入出力コントローラが、M個のそれぞれのデータに対するM個のライトトランザクションを開始する。

M個のデータのうちの第Iデータ(Iは、 $I=1, 2, \dots, M$ を満たす整数の何れか)は、複数のプロセッサノードのうちの第1プロセッサノードをホームとするデータである。第Iライトメッセージは第1プロセッサノードの主記憶部に格納された複数のデータのうちの第Iデータの値を第Iライトメッセージで

指定される値に更新するための命令である。

メモリアクセス処理方法は、第1入出力コントローラが、第Iライトトランザクションの処理として、第I書き込み要求メッセージを第1プロセッサノードに出力するステップを含む。

メモリアクセス処理方法はまた、第1プロセッサノードのメモリコントローラが、第I書き込み要求メッセージを受け取ったとき、第Iデータに対して、フリー状態情報に代えて、第Iデータに対するプロセッサや入出力デバイスからの読み出し要求や他の書き込み要求メッセージを受け付けることができないライトロック状態情報を第1プロセッサノードのディレクトリに格納し、第I書き込み要求メッセージに対して第I書き込み許可メッセージを第1入出力コントローラに出力するステップを含む。

メモリアクセス処理方法は更に、第1入出力コントローラが、第I書き込み許可メッセージを受け取ったとき、第Iライトトランザクションの処理として更新メッセージ発行処理を行うステップを含む。

第1入出力コントローラが行う更新メッセージ発行処理は、第1から第I書き込み許可メッセージまでのI個の書き込み許可メッセージを既に受け取っているか否かを検査するステップを含む。

更新メッセージ発行処理はまた、I個の書き込み許可メッセージをまだ受け取っていないければ、第I更新メッセージ発行処理を終了するステップと、I個の書き込み許可メッセージを既に受け取っていれば、第Iライトメッセージで指定される値を含む第I更新メッセージを第1プロセッサノードに出力して第Iライトトランザクションを完了させ、 $(I+1)$ がM以下であれば第 $(I+1)$ ライトトランザクションの更新メッセージ発行処理を行い、 $(I+1)$ がMより大きければ第I更新メッセージ発行処理を終了するステップとを含む。

メモリアクセス処理方法は更に、第1プロセッサノードのメモリコントローラが、第I更新メッセージを受け取ったとき、第Iデータに対してライトロック状態情報に代えてフリー状態情報を第1プロセッサノードのディレクトリに格納すると共に、第1プロセッサノードの主記憶部に格納された第Iデータの値を第I更新メッセージで指定される値に更新するステップを含む。

[第10の態様]

第9の態様によるメモリアクセス処理方法において、メモリアクセス処理方法は更に、複数の入出力ノードの各々の複数の入出力デバイスによって発行されるM個のライトメッセージを調停するステップを含む。

[第11の態様]

第9又は第10の態様によるメモリアクセス処理方法において、メモリアクセス処理方法は更に、第Iデータに対してライトロック状態情報が第1プロセッサノードのディレクトリに格納されているときに、第1プロセッサノードのメモリコントローラが、第Iデータに対する第I書き込み要求メッセージを第1入出力コントローラから受け取った場合、第I書き込み要求メッセージに対して第I開放要求メッセージを第1入出力コントローラに出力するステップを含む。メモリアクセス処理方法は更に、第1入出力コントローラが、第I開放要求メッセージを受けて第I書き込み要求メッセージを第1プロセッサノードのメモリコントローラに出力すると共に、開放処理を行なうステップを含む。

開放処理を行なうステップは、第Iデータに後続する第Kライトトランザクション {Kは、 $K = I + 1, I + 2, \dots, M$ を満たす整数、 $I + 1$ は、 $I < (I + 1) < M$ を満たす整数、 $I + 2$ は、 $(I + 1) < (I + 2) < M$ を満たす整数}の進捗を検査し、未だ第K書き込み要求メッセージを発行していない場合、第K書き込み要求メッセージの発行を停止するステップを含む。

開放処理を行なうステップはまた、既に第K書き込み要求メッセージを発行し第K書き込み許可メッセージを受け取っている場合、第K開放メッセージを第Kデータのホームである第2プロセッサノードのメモリコントローラに出力するステップと、既に第K書き込み要求メッセージを発行しまだ第K書き込み許可メッセージを受け取っていない場合は、第K書き込み許可メッセージを受け取った時点で第K開放メッセージの発行を行うステップとを含む。

メモリアクセス処理方法は更に、第2プロセッサノードのメモリコントローラが、第K開放メッセージを受け取ったとき、第Iデータに対してライトロック状態情報に代えてフリー状態情報を第2プロセッサノードのディレクトリに格納するステップを含む。

[第12の態様]

第9又は第10の態様によるメモリアクセス処理方法において、ディレクトリに格納される状態情報は、さらに割り込み可ライトロック状態情報とリクエストロック状態情報とを含む。

メモリアクセス処理方法は更に、第1入出力コントローラが第I書き込み要求メッセージを発行する際に、第1から第(I-1)書き込み許可メッセージまでの(I-1)個の書き込み許可メッセージを既に受け取っているか否かを検査し、受け取り済みであれば第I強書き込み要求メッセージを、受け取り済みでなければ第I弱書き込み要求メッセージを発行するステップを含む。

メモリアクセス処理方法は更に、第1プロセッサノードのディレクトリに格納されている第Iデータの状态情報がフリー状態情報であるときに、第1プロセッサノードのメモリコントローラが、第I強書き込み要求メッセージを第1入出力コントローラから受け取った場合、フリー状態情報に代えて、プロセッサや入出力デバイスからの読み出し要求や、第Iデータに対する他の強・弱書き込み要求メッセージ両方を受け付けることができないライトロック状態情報を第1プロセッサノードのディレクトリに格納し、第I書き込み許可メッセージを第1入出力コントローラに出力するステップを含む。

メモリアクセス処理方法は更に、第1プロセッサノードのディレクトリに格納されている第Iデータの状态情報がフリー状態情報であるときに、第1プロセッサノードのメモリコントローラが、第I弱書き込み要求メッセージを第1入出力コントローラから受け取った場合、フリー状態情報に代えて、プロセッサや入出力デバイスからの読み出し要求や、第Iデータに対する他の弱書き込み要求メッセージを受け付けることができない割り込み可ライトロック状態情報を第1プロセッサノードのディレクトリに格納し、第I書き込み許可メッセージを第1入出力コントローラに出力するステップを含む。

割り込み可ライトロック状態情報は、この状態にした弱書き込み要求メッセージを発行した入出力コントローラを特定する情報を含む。

メモリアクセス処理方法は更に、第1プロセッサノードのディレクトリに格納されている第Iデータの状态情報がライトロック状態情報あるいはリクエストロ

ック状態情報であるときに、第1プロセッサノードのメモリコントローラが、第I強書き込み要求メッセージを第1入出力コントローラから受け取った場合、第I不許可メッセージを第1入出力コントローラに出力するステップを含む。

メモリアクセス処理方法は更に、第1プロセッサノードのディレクトリに格納されている第Iデータの状态情報が割り込み可ライトロック状態情報、ライトロック状態情報、あるいはリクエストロック状態情報であるときに、第1プロセッサノードのメモリコントローラが、第I弱書き込み要求メッセージを第1入出力コントローラから受け取った場合、第I不許可メッセージを第1入出力コントローラに出力するステップを含む。

メモリアクセス処理方法は更に、第1入出力コントローラが、第I不許可メッセージを受け取ったとき、第1から第(I-1)書き込み許可メッセージまでの(I-1)個の書き込み許可メッセージを既に受け取っているか否かを検査し、受け取り済みであれば第I強書き込み要求メッセージを、受け取り済みでなければ第I弱書き込み要求メッセージを発行するステップを含む。

メモリアクセス処理方法は更に、第1プロセッサノードのディレクトリに格納されている第Iデータの状态情報が割り込み可ライトロック状態情報であるときに、第1プロセッサノードのメモリコントローラが、第I強書き込み要求メッセージを第1入出力コントローラから受け取った場合、割り込み可ライトロック状態情報に代えて、プロセッサや入出力デバイスからの読み出し要求や、第Iデータに対する他の弱書き込み要求メッセージを受け付けることができないリクエストロック状態情報を第1プロセッサノードのディレクトリに格納し、第I再試行要求メッセージをディレクトリに格納された入出力コントローラを特定する情報が指す第2入出力コントローラに宛てて出力するステップを含む。

メモリアクセス処理方法は更に、第I再試行要求メッセージを受け取った第2入出力コントローラが、再試行処理を行うステップを含む。

[第13の態様]

第12の態様によるメモリアクセス処理方法において、第2入出力コントローラが行う再試行処理は、第I開放メッセージを第1プロセッサノードのメモリコントローラに出力するステップを含む。

再試行処理はまた、第 I データとアドレスを同じくするライトトランザクションで書き込み許可メッセージを受け取り済みのものに関して、まだ更新メッセージを発行していなければ更新メッセージの発行を停止して、第 I 開放メッセージの発行後に、第 1 プロセッサノードに宛てて書き込み要求メッセージを発行するステップを含む。

再試行処理は更に、第 I 開放メッセージを受けた第 1 プロセッサノードのメモリコントローラが、リクエストロック状態情報に代えてライトロック状態情報を第 1 プロセッサノードのディレクトリに格納し、第 I 書き込み許可メッセージを第 1 入出力コントローラに宛てて出力するステップを含む。

請 求 の 範 囲

1. ネットワークに接続された複数の資源ノードと前記ネットワークに接続された複数の制御ノードと複数の制御コントローラとを具備するプロセッサシステムにおいて、

前記複数の資源ノードはそれぞれ、複数の資源を保持する資源保持部と、前記複数の資源がフリー状態にあるかロック状態にあるかを管理する資源情報保持部とを含み、

前記複数の制御ノードはそれぞれ、前記資源に対する要求を複数発行する複数の制御デバイスを含み、前記複数の要求を発行した制御デバイスを含む制御ノードは前記複数の要求を対応する制御コントローラに転送し、

前記要求を受けた少なくとも1つの制御コントローラは、該要求が対象とする資源を保持する資源ノードに要求メッセージを発行し、

前記要求メッセージを受けた資源ノードは、対象とする資源の状態がフリー状態にあることを確認して前記資源の状態をロック状態に変更し、続いて許可メッセージを前記少なくとも1つの制御コントローラに発行し、

前記許可メッセージを受けた前記少なくとも1つの制御コントローラは、当該少なくとも1つの制御コントローラが受けた前記要求に先行する他の複数の要求について全て許可メッセージを受け取り済みであることを確認して、前記要求が対象としている資源を保持する資源ノードに更新メッセージを発行し、

前記更新メッセージを受けた資源ノードは、対象とする前記資源の状態をフリー状態に変更する、

ことを特徴とするプロセッサシステム。

2. 請求項1に記載のプロセッサシステムにおいて、

前記複数の制御コントローラはそれぞれ、前記複数の制御ノード内に備えられ、各制御ノードの制御コントローラは、各制御ノードの複数の制御デバイスによって発行される前記複数の要求を調停するセレクタを含むことを特徴とするプロセッサシステム。

3. 請求項1に記載のプロセッサシステムにおいて、
前記複数の制御コントローラは前記ネットワークに接続され、
各制御ノードは、前記複数の制御ノードの各々の前記複数の制御デバイスによって発行される前記複数の要求を調停するセクタを含むことを特徴とするプロセッサシステム。

4. 請求項1に記載のプロセッサシステムにおいて、
前記複数の制御コントローラはそれぞれ、前記複数の資源ノード内に備えられ、
各制御ノードは、前記複数の制御ノードの各々の前記複数の制御デバイスによって発行される前記複数の要求を調停するセクタを含むことを特徴とするプロセッサシステム。

5. 請求項1に記載のプロセッサシステムにおいて、
前記複数の資源ノードと前記複数の制御ノードとは、それぞれ複数のノードを構成し、

各ノードは、前記複数のノードの各々の前記複数の制御デバイスによって発行される前記複数の要求を調停するセクタを含むことを特徴とするプロセッサシステム。

6. 請求項1～5のいずれかに記載のプロセッサシステムにおいて、
各制御コントローラは開放処理部を具備し、
前記要求メッセージを受けた前記資源ノードは、前記対象とする資源の状態がロック状態にあった場合、開放要求メッセージを前記少なくとも1つの制御コントローラに発行し、

前記開放要求メッセージを受けた前記少なくとも1つの制御コントローラの開放処理部は開放処理を開始し、

前記開放処理は、

該要求が対象としている資源を保持する資源ノードに要求メッセージを発行する処理と、

新規要求の受付を停止する処理と、

該少なくとも1つの制御コントローラが受けた前記要求に後続する他の複数の後続要求の各々について、既に許可メッセージを受け取っていれば、あるいは開

放処理中に許可メッセージを受け取れば、該後続要求が対象としている資源を保持する資源ノードに開放メッセージを発行する処理とを含み、

前記開放要求メッセージの要求に対する許可メッセージを受け取った前記少なくとも1つの制御コントローラは開放処理を終了することを特徴とするプロセッサシステム。

7. 請求項1～5のいずれかに記載のプロセッサシステムにおいて、

前記資源情報保持部が管理する状態は、割り込み可ロック状態とリクエストロック状態とを含み、

前記要求を受けた前記少なくとも1つの制御コントローラは、該少なくとも1つの制御コントローラが受けた前記要求に先行する他の複数の要求の全てについて許可メッセージを受け取り済みでなければ弱要求メッセージを発行し、

前記弱要求メッセージを受けた前記資源ノードは、

前記対象とする資源の状態がフリー状態にあることを確認して、前記資源の状態を割り込み可ロック状態に変更し、続いて許可メッセージを前記少なくとも1つの制御コントローラに発行し、

前記割り込み可ロック状態は、前記少なくとも1つの制御コントローラを特定する情報を含み、

前記要求メッセージを受けた前記資源ノードは、前記対象とする資源の状態がロック状態にあった場合、不許可メッセージを前記少なくとも1つの制御コントローラに発行し、

前記弱要求メッセージを受けた前記資源ノードは、前記対象とする資源の状態がロック状態あるいは割り込み可ロック状態にあった場合、不許可メッセージを前記少なくとも1つの制御コントローラに発行し、

前記不許可メッセージを受けた前記少なくとも1つの制御コントローラは、該少なくとも1つの制御コントローラが受けた前記要求に先行する他の複数の要求の全てについて許可メッセージを受け取り済みであれば要求メッセージを、受け取り済みでなければ弱要求メッセージを発行し、

前記要求メッセージを受けた前記資源ノードは、前記対象とする資源の状態が割り込み可ロック状態にあった場合、前記資源の状態をリクエストロック状態に

変更し、続いて前記割り込み可ロック状態に含まれる情報が示す制御コントローラに宛てて再試行要求メッセージを出力し、

前記再試行要求メッセージを受け取った前記制御コントローラは、再試行処理を行い、

前記再試行処理は、

該資源に対する要求を特定する処理と、

該要求が対象としている資源を保持する資源ノードに開放メッセージを発行する処理と、

特定した要求がまだ更新メッセージを発行していなければ、該要求について許可メッセージをまだ受け取っていない状態に変更し、該制御コントローラが受けた前記要求に先行する他の複数の要求の全てについて許可メッセージを受け取り済みであれば要求メッセージを、受け取り済みでなければ弱要求メッセージを発行する処理とを含み、

前記開放メッセージを受けた前記資源ノードは、前記資源の状態をロック状態に変更し、続いて許可メッセージを前記少なくとも1つの制御コントローラに発行することを特徴とするプロセッサシステム。

8. 請求項1に記載のプロセッサシステムにおいて、

前記資源ノードは複数のプロセッサを有するプロセッサノードであり、

該プロセッサノードは、前記資源保持部、前記資源情報保持部として主記憶部、ディレクトリをそれぞれ備えるとともに、前記複数のプロセッサと前記主記憶部及びディレクトリに接続されたメモリコントローラを備え、

前記制御ノードは、前記複数の制御デバイスとして複数の入出力デバイスを含む入出力ノードであり、

前記制御コントローラは入出力コントローラであることを特徴とするプロセッサシステム。

9. ネットワークに接続された複数の資源ノードと前記ネットワークに接続された複数の制御ノードと複数の制御コントローラとを具備するプロセッサシステムであって、前記複数の資源ノードがそれぞれ、複数の資源を保持する資源保持部と、前記複数の資源がフリー状態にあるかロック状態にあるかを管理する資

源情報保持部とを含み、前記複数の制御ノードがそれぞれ、前記資源に対する要求を複数発行する複数の制御デバイスを含む、プロセッサシステムに適用されるアクセス処理方法であって、

前記複数の要求を対応する少なくとも1つの制御コントローラに転送するステップと、

該要求が対象とする資源を保持する資源ノードに要求メッセージを発行するステップと、

対象とする資源の状態がフリー状態にあることを確認して前記資源の状態をロック状態に変更し、続いて許可メッセージを前記少なくとも1つの制御コントローラに発行するステップと、

当該少なくとも1つの制御コントローラが受けた前記要求に先行する他の複数の要求について全て許可メッセージを受け取り済みであることを確認して、前記要求が対象としている資源を保持する資源ノードに更新メッセージを発行するステップと、

対象とする前記資源の状態をフリー状態に変更するステップと、
を含むことを特徴とするアクセス処理方法。

10. 請求項9に記載のアクセス処理方法において、

更に、各制御ノードの複数の制御デバイスによって発行される前記複数の要求を調停するステップを含むことを特徴とするアクセス処理方法。

11. 請求項9に記載のアクセス処理方法において、

更に、前記複数の資源ノードと前記複数の制御ノードとを、それぞれ複数のノードとして構成するステップと、

前記複数のノードの各々の前記複数の制御デバイスによって発行される前記複数の要求を調停するステップとを含むことを特徴とするアクセス処理方法。

12. 請求項9～11のいずれかに記載のアクセス処理方法において、

更に、前記対象とする資源の状態がロック状態にあった場合、開放要求メッセージを前記少なくとも1つの制御コントローラに発行するステップと、

前記開放要求メッセージを受けた前記少なくとも1つの制御コントローラが開放処理を開始するステップとを含み、

前記開放処理は、
該要求が対象としている資源を保持する資源ノードに要求メッセージを発行するステップと、

新規要求の受付を停止するステップと、

該少なくとも1つの制御コントローラが受けた前記要求に後続する他の複数の後続要求の各々について、既に許可メッセージを受け取っていれば、あるいは開放処理中に許可メッセージを受け取れば、該後続要求が対象としている資源を保持する資源ノードに開放メッセージを発行するステップとを含み、

アクセス処理方法は更に、前記開放要求メッセージの要求に対する許可メッセージを受け取った前記少なくとも1つの制御コントローラは開放処理を終了するステップを含むことを特徴とするアクセス処理方法。

13. 請求項9～11のいずれかに記載のアクセス処理方法において、

前記資源情報保持部が管理する状態は、割り込み可ロック状態とリクエストロック状態とを含み、

アクセス処理方法は更に、前記少なくとも1つの制御コントローラが受けた前記要求に先行する他の複数の要求の全てについて許可メッセージを受け取り済みでなければ弱要求メッセージを発行するステップと、

前記対象とする資源の状態がフリー状態にあることを確認して、前記資源の状態を割り込み可ロック状態に変更し、続いて許可メッセージを前記少なくとも1つの制御コントローラに発行するステップとを含み、

前記割り込み可ロック状態は、前記少なくとも1つの制御コントローラを特定する情報を含み、

アクセス処理方法は更に、前記対象とする資源の状態がロック状態にあった場合、不許可メッセージを前記少なくとも1つの制御コントローラに発行するステップと、

前記対象とする資源の状態がロック状態あるいは割り込み可ロック状態にあった場合、不許可メッセージを前記少なくとも1つの制御コントローラに発行するステップと、

前記少なくとも1つの制御コントローラが受けた前記要求に先行する他の複数の

の要求の全てについて許可メッセージを受け取り済みであれば要求メッセージを、受け取り済みでなければ弱要求メッセージを発行するステップと、

前記対象とする資源の状態が割り込み可ロック状態にあった場合、前記資源の状態をリクエストロック状態に変更し、続いて前記割り込み可ロック状態に含まれる情報が示す制御コントローラに宛てて再試行要求メッセージを出力するステップと、

再試行処理を行うステップとを含み、

前記再試行処理は、

該資源に対する要求を特定するステップと、

該要求が対象としている資源を保持する資源ノードに開放メッセージを発行するステップと、

特定した要求がまだ更新メッセージを発行していなければ、該要求について許可メッセージをまだ受け取っていない状態に変更し、該制御コントローラが受けた前記要求に先行する他の複数の要求の全てについて許可メッセージを受け取り済みであれば要求メッセージを、受け取り済みでなければ弱要求メッセージを発行するステップとを含み、

アクセス処理方法は更に、前記資源の状態をロック状態に変更し、続いて許可メッセージを前記少なくとも1つの制御コントローラに発行するステップを含むことを特徴とするアクセス処理方法。

14. 請求項9に記載のアクセス処理方法において、

前記資源ノードは複数のプロセッサを有するプロセッサノードであり、

該プロセッサノードは、前記資源保持部、前記資源情報保持部として主記憶部、ディレクトリをそれぞれ備えるとともに、前記複数のプロセッサと前記主記憶部及びディレクトリに接続されたメモリコントローラを備え、

前記制御ノードは、前記複数の制御デバイスとして複数の入出力デバイスを含む入出力ノードであり、

前記制御コントローラは入出力コントローラであることを特徴とするプロセッサシステム。

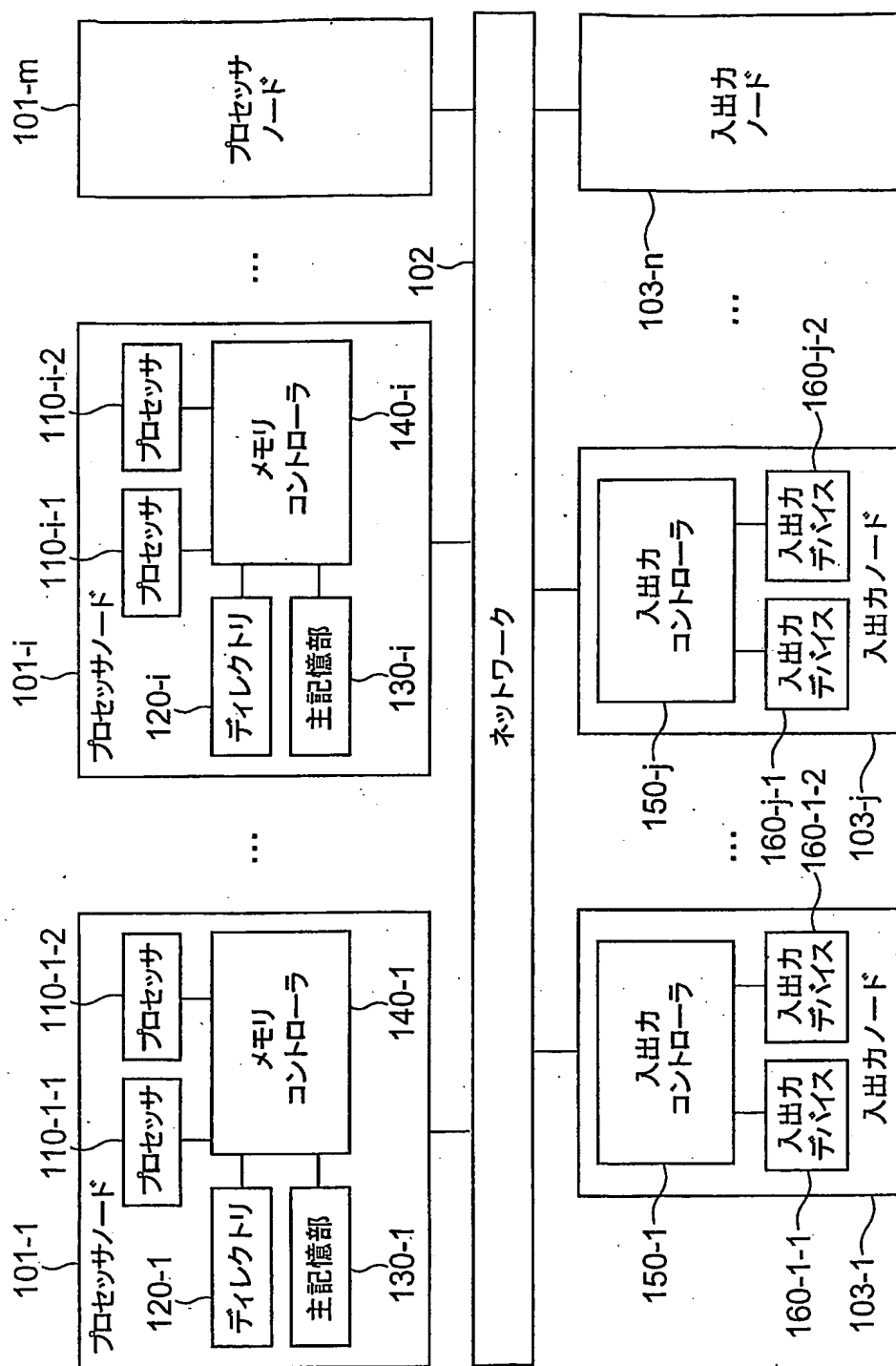


図 1

[illegible]

2
✕

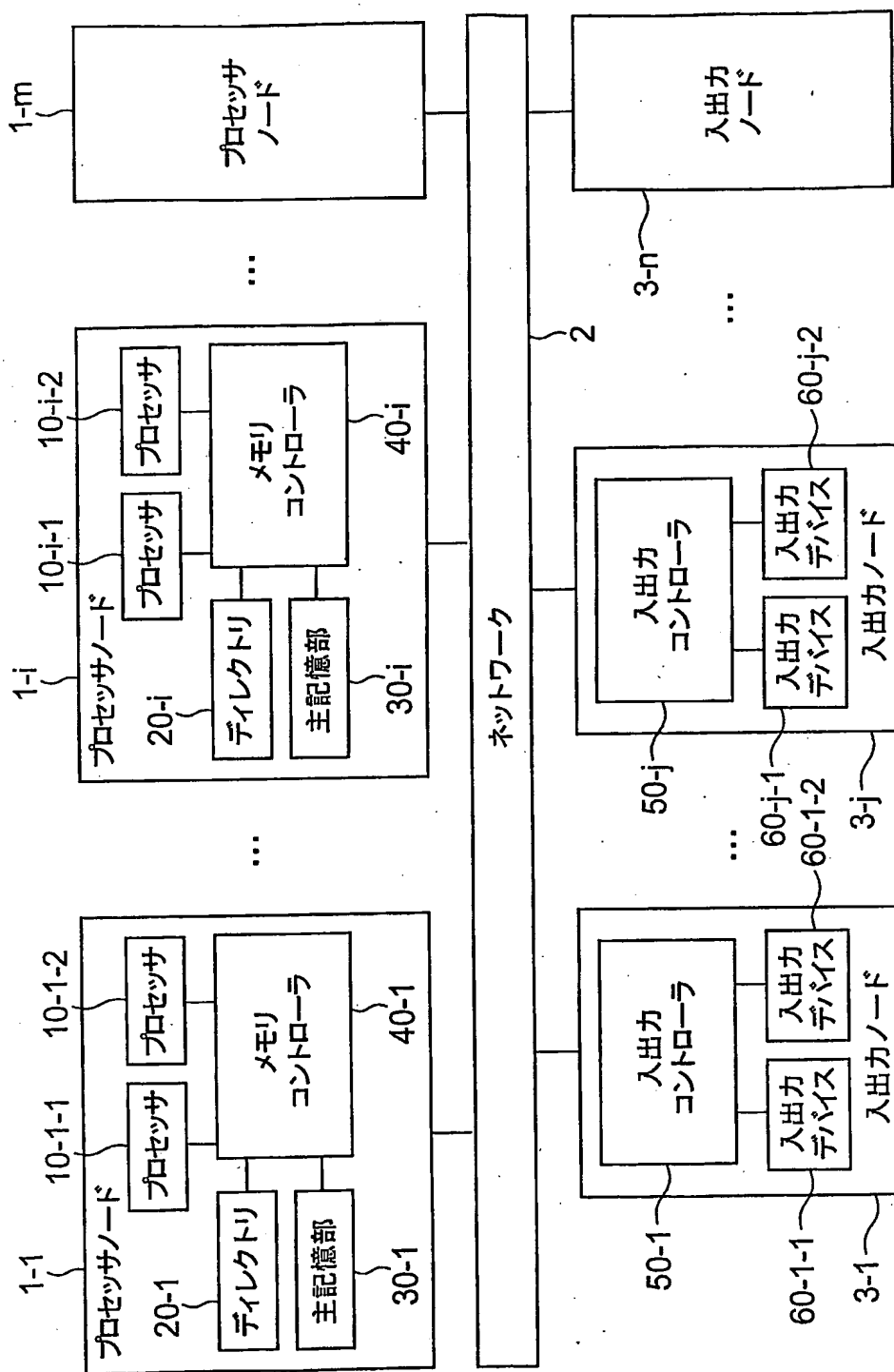


図 4

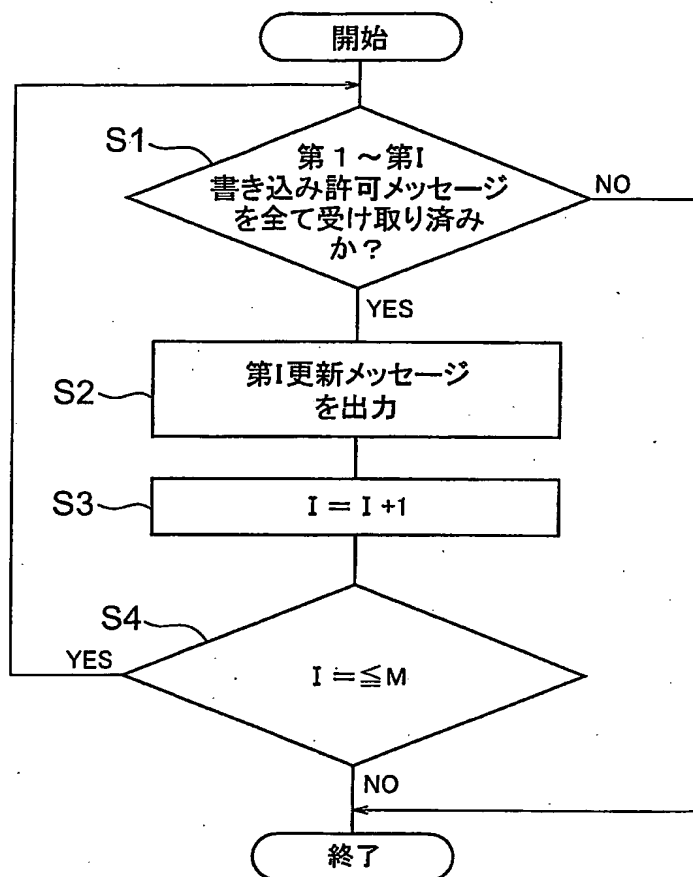


図 5

[illegible]

60-1-1:入出力デバイス

50-1:入出カコントロール

2: ネットワーク

40-1:メモリコントローラ
ディレクトリ20-1に
格納されたデータAの状態

ディレクトリ20-1に格納されたデータBの状態

40-2:メモリコントローラ

ディレクトリ20-2に
格納されたデータCの状態

6. 凶

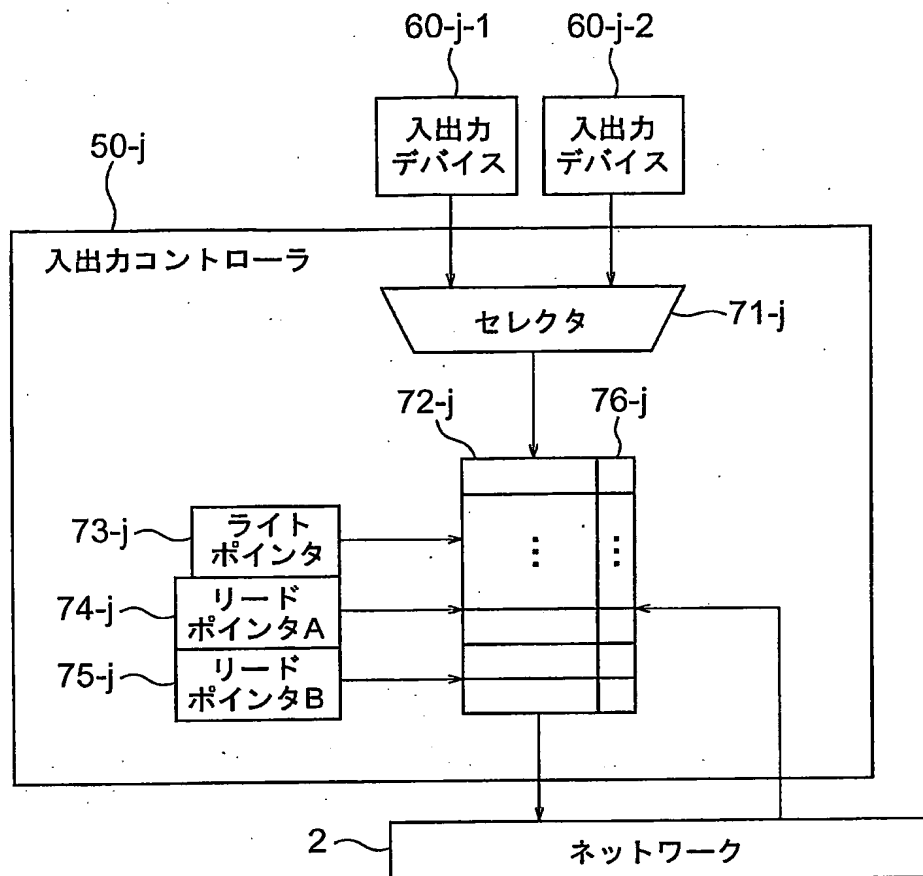


図 7

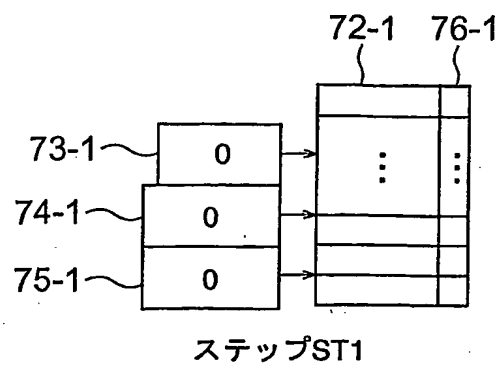


図 8A

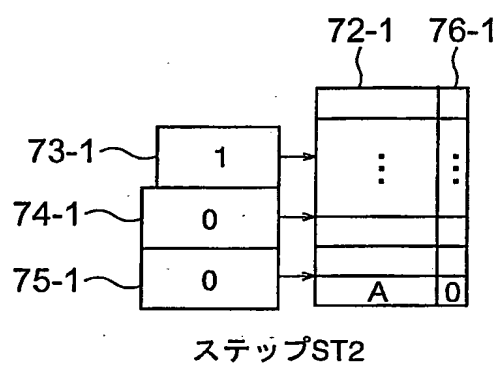


図 8B

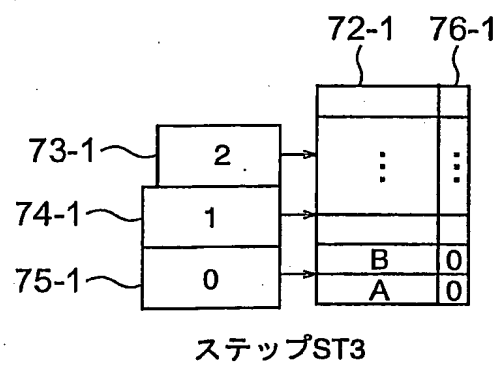


図 8C

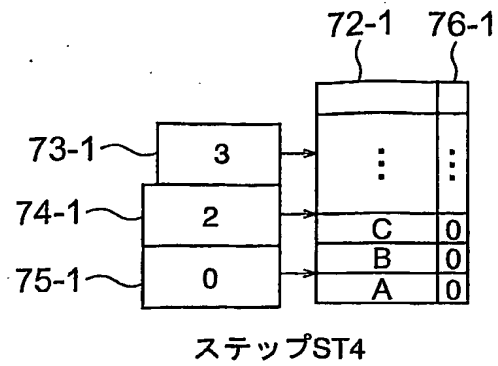


図 8D

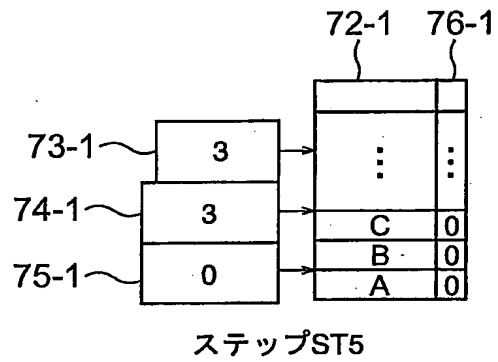


図 8E

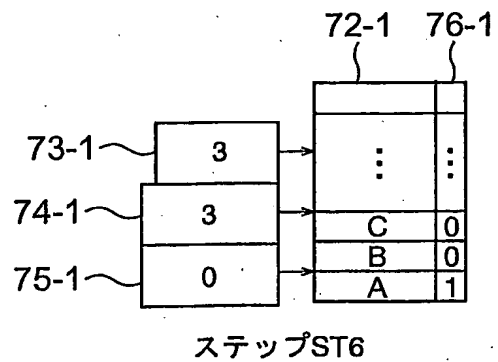
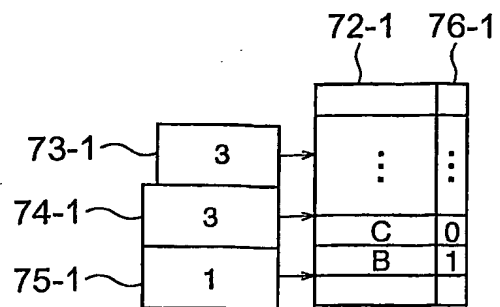
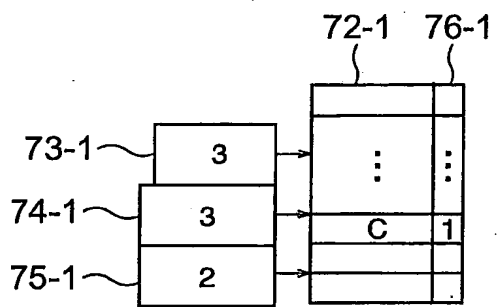


図 8F



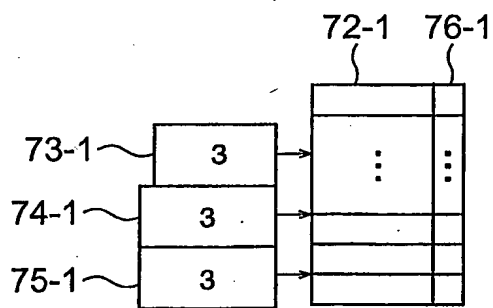
ステップST7

図 8G



ステップST8

図 8H



ステップST9

図 8I

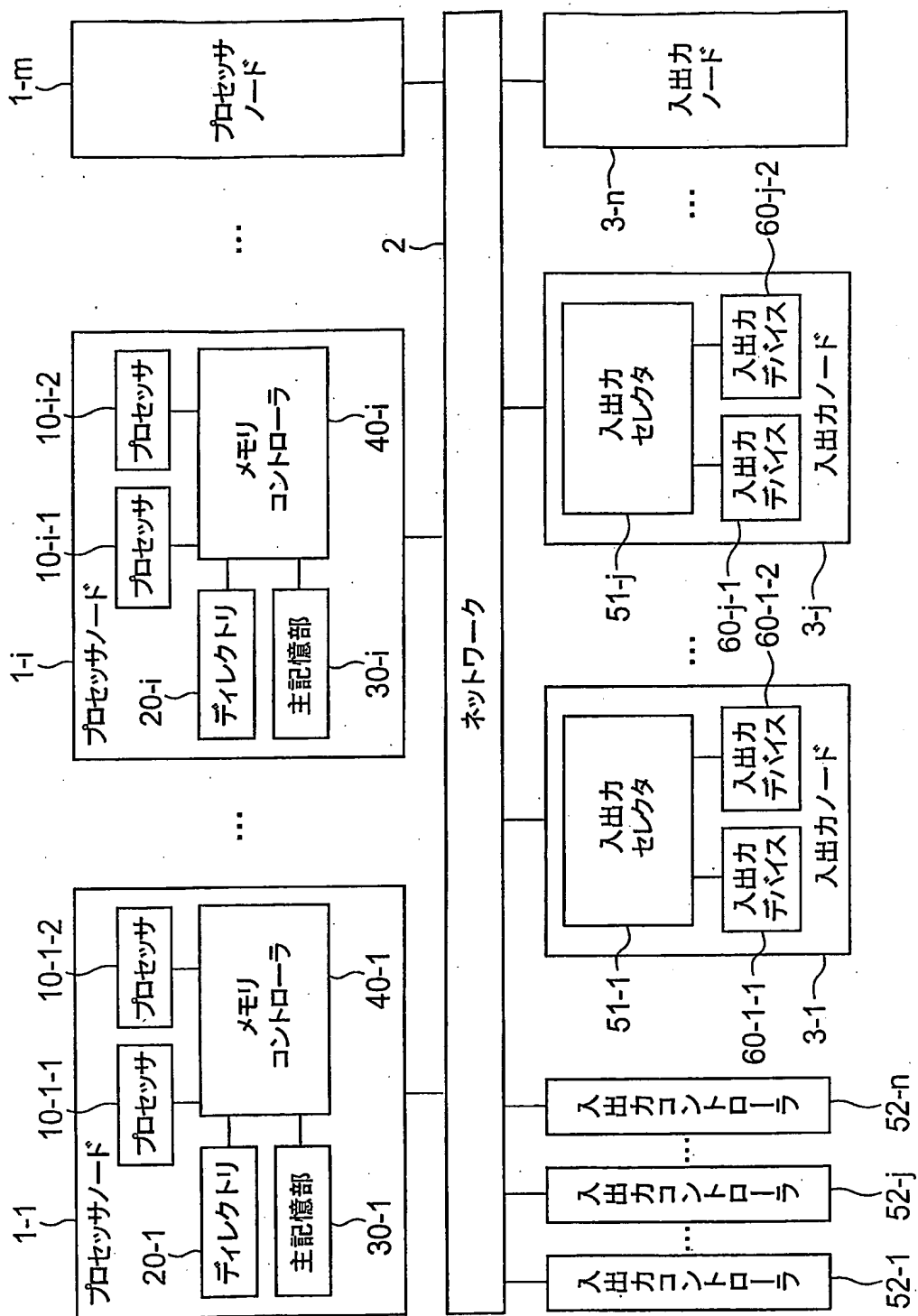


図 9

	1	2	3	4	5	6	7	8	9
60-1-1: 入出力デバイス	ライト A								
50-1: 入出力コントローラ		ライト A				許可 A			
2:ネットワーク			要求 A		許可 A		更新 A		
40-1: メモリコントローラ				要求 A				更新 A	
20-1:ディレクタトリ									
			U,000			W,000			U,000

図10

	1	2	3	4	5	6	7	8	9	10	11	12	13
60-1-1: 入出力デバイス	ライト A												
50-1: 入出力コントローラ		ライト A								許可 A			
2:ネットワーク			要求 A		書戻 要求A		書戻 A		許可 A		更新 A		
40-1: メモリコントローラ				要求 A				書戻 A				更新 A	
20-1:ディレクトリ	D,010			R,010			W,000			U,000			
40-2: メモリコントローラ						書戻 要求A							

図12

	1	2	3	4	5	6	7
60-1-1: 入出力デバイス	ライト A						
		ライト A				不許可 A	
50-1: 入出力コントローラ							要求 A
			要求 A		不許可 A		
2: ネットワーク							
40-1: メモリコントローラ				要求 A			
20-1: ディレクトリ							
	R, --- or W, 000						

図13

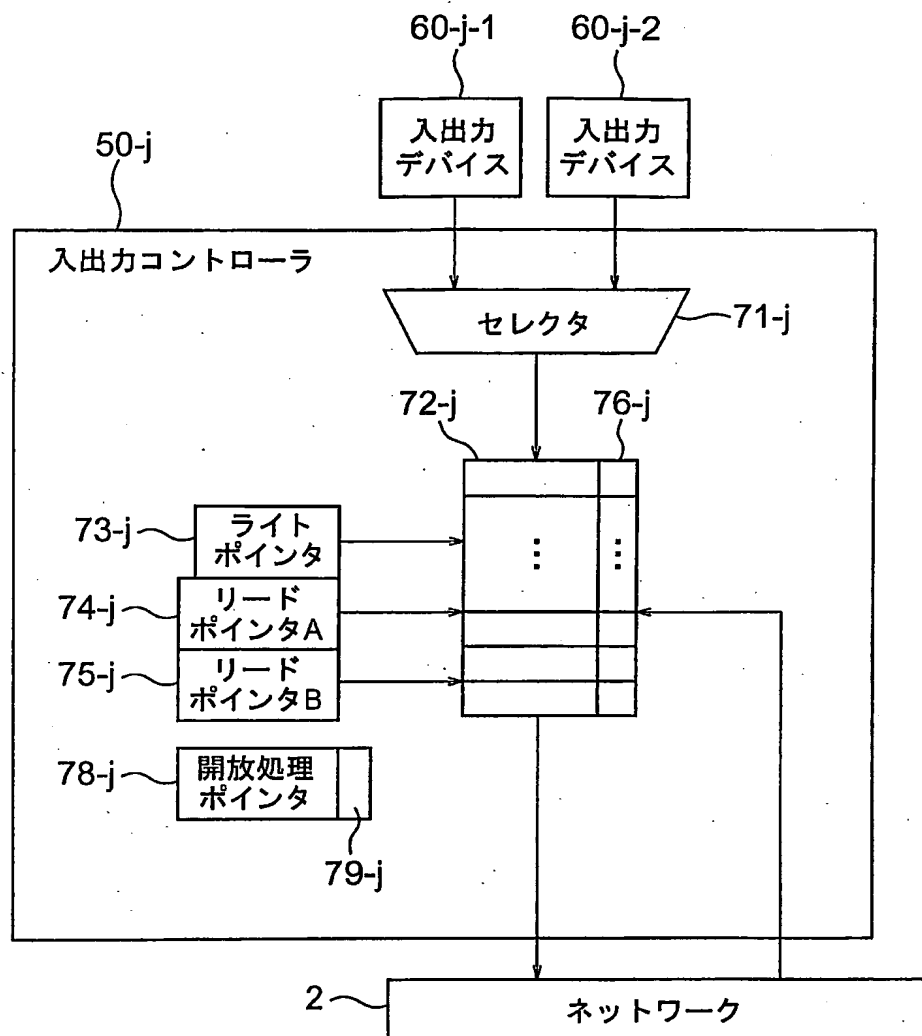


図14